

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-171510

(43)Date of publication of application : 30.06.1997

(51)Int.Cl.

G06F 17/27

G06F 17/21

(21)Application number : 08-297425

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 18.10.1996

(72)Inventor : HAYASHI KOICHI

(30)Priority

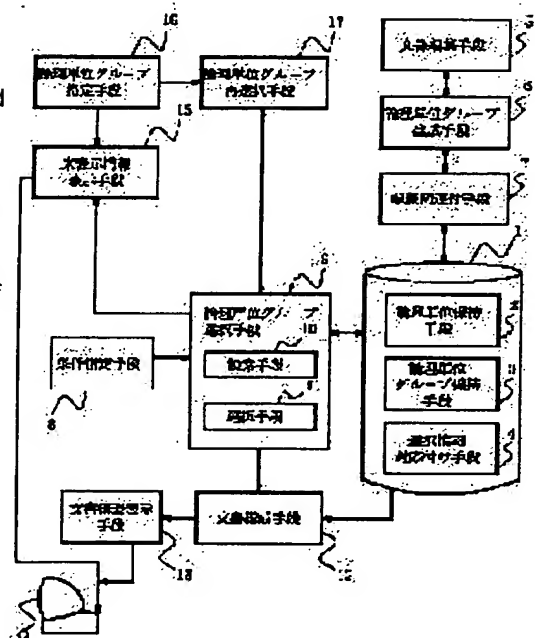
Priority number : 07296154 Priority date : 19.10.1995 Priority country : JP

## 54) STRUCTURED DATA PROCESSOR

57)Abstract:

**PROBLEM TO BE SOLVED:** To realize a cooperative writing smoothly by enabling the restoration of a past document and the preparation of the document which does not exist in the past even if the document is freely edited by a writer without restricting the writer by editing.

**SOLUTION:** In a word processor for which a structured data processing is applied to structured document data, the logical unit including at least either one of the contents part being a part of the contents of the structure document consisting of a tree structure or the anchor showing the related position with the structure of a subordinate hierarchy is held by a logical unit holding means 2, and the stream consisting of the logical unit located at the same hierarchy of the tree structure constituting the structure document is held by a logical unit group holding means 3. When the document structure is edited by a document editing means 5, a logical unit group generation means 6 newly generates the stream corresponding to the editing of the document and a reference relating means 7 relates the newly generated stream and a prescribed anchor. As a result, the newly generated stream is incorporated into an existing structured document and the structured document according to the edition of the document is prepared.



Copyright (C); 2003 Japan Patent Office

## BEST AVAILABLE COPY

PAGE BLANK (USPTO)

\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

## DETAILED DESCRIPTION

---

### Detailed Description of the Invention]

0001]

Field of the Invention] Even if this invention allows editing freely about the processor and the approach creation, edit, reservation, etc. carry out the structured data which constitutes the graph structure called document data etc., without giving a writer the constraint on data editing especially, it relates to the structured data processor and approach of performing restoration and co-authoring of the past structured data without trouble.

0002]

Description of the Prior Art] Generally, when updating is repeated by settlement of the document information of a file etc., the technique of guaranteeing taking out the document of the condition of the past corresponding to specific conditions is called version control, and the condition of the past document is called a version. In addition to the use that the document of the past condition can be restored, such version control has the use that the co-authoring activity by two or more persons in charge is supportable etc.

0003] Here, in the former, when version control made a settlement of information, such as a file, a unit, and was performed and joint edit of the file was carried out, management of restricting so that two or more persons in charge cannot edit the same file simultaneously was performed. However, it is more convenient to edit a document in partial units, such as a chapter and a knot, when writing a document with dramatically big size, or when writing a document by two or more persons in charge. Then, invention which made possible \*\* of co-authoring by two or more persons in charge is made by structuring a document, managing the version for every part, and fixing, or adding a limit so that the same part cannot be simultaneously edited by two or more persons in charge so that an assignment may be opted for and written by each person in charge person.

0004] The version control of the structured document in the former is explained in more detail with reference to drawing 84 - drawing 89. The structure of a document is expressed as the tree structure with sequence, the node showing partial units, such as a chapter and a knot, is structured using the link which shows a parentage, and the node which included the content (the content of data of partial units, such as a chapter and a knot) in the end of this tree structure is connected. That is, in the example of the document structure shown in drawing 84, if Node A is made into the document root, a chapter and nodes C and D will correspond to a knot, and Node B will serve as a content block in which the node (u) of an end, (v), and (x) include the contents of data, such as a text. If the whole document structure changes, the unit of version control will stop in addition, being stabilized by them since partial units, such as a chapter and a knot, are defined by the structure as the whole document. So, in the version control of the document structure in such the former, the whole document structure is fixed and a version control line is about a content block.

0005] And in performing edit processing in which create a new content block and it changes for the existing content block, as shown in drawing 85, while replacing a new content block (y) with the existing content block (x), an old content block (x) is matched with the original location, and carries out a storage management. By this, when restoring the condition (namely, condition before edit) of the past of the document concerned, the desired end can be attained only by replacing an old content block (x) with a new content block (y) again.

0006] In such version control, since it described above, when the whole document structure is changed, this cannot be managed and the past version cannot be restored only by replacing a content block old as mentioned above with a new content block. For example, when the new substructure (node) E which adds the content block (z) of a new partial unit, and corresponds to drawing 86 so that it may be shown is inserted in document structure to the original document shown in drawing 85, When the substructure (node) D which starts a part for a content block (x) as shown in drawing 87 is deleted, as shown in drawing 88, when the sequence of Substructure (node) C and Substructure (node) D is replaced, Since the

**BEST AVAILABLE COPY**

PAGE BLANK (USPTO)



whole document structure is changed when the location of the substructure (node) D concerning a part for a content block (x) is moved, as shown in drawing 89, the document of the past condition cannot be restored only by replacing a content block.

[0007] In addition, the document preparation system which supports co-authoring to the same document is indicated by JP,6-35914,A using the version control of a content block as shown in drawing 85. In this document preparation system, when the content block of the same node is edited by two or more writing persons in charge, it is made to correspond to the content block which each person in charge edited, a version node is created, and each content block is managed by matching with a version node. Furthermore, in case the access privilege is set to each version node and a writing person in charge edits the content block in document structure, based on a version node, the content block to which the access privilege which suits the person in charge is set is shown, and the edit is enabled.

[0008] moreover, to JP,7-44563,A or JP,6-131343,A Express the structure of a document by the tree structure which consists of a node and a link, and when logical structure relation is the same about two or more parts in a document from which the content differs mutually By compounding the logical structure of these parts, holding as one compound statement document structure, and giving further the node which multiplexed the attribute for distinguishing each substructure by multiplexing a node The multiplex document processing system system which can choose the tree structure corresponding to required document structure out of two or more tree structures is indicated. According to this multiplex document processing system system, it can express flexibly and the tree structure corresponding to required document structure can be further chosen out of two or more tree structures if needed rather than it divides document structure into whole structure and the content simply.

[0009] A version control technique is a technique used in various fields, such as a software development and a CAD database, besides edit of document data here, and the version control about the complicated data whole structure where each component (namely, partial data) had a version is an important technical problem also in a general version control technique. As a conventional technique about such version control, various following techniques are known, for example.

[0010] The version of the whole structure mentioned above is called a configuration, and the technique of managing the version of components independently is shown by "R.Katz, 'Toward a Unified Framework for Version Modeling in Engineering Databases', ACM Computing Surveys, vol.22, no.4, and 1990" which described comprehensively the version control technique of a CAD database with complicated structure. Although various version control techniques are shown in this reference, it is that, as for the object of version control, all restore the past condition. The approach using a derivation tree as an approach for that and the approach using a layer show. The approach using a derivation tree restores the past condition by managing a derivation tree (derivation tree) by the relation which version was derived from the thing of which version as hysteresis of modification, and choosing one condition from the inside. Thus, storage capacity required for implementation of version control can be made to mitigate by recording only the difference from the condition in front of one using a derivation tree. In addition, "Sten Minor & Boris Magnusson, 'A Model for Semi-(a) Synchronous Collaborative Editing, proceedings of ECSCW'93, pp.219-231, and 1993" show the example which applied the version control of the whole structure by the derivation tree to document edit. [0011] When it changes, the generated components and the updated components are held as a settlement called a layer, and it enables it to express the condition of having existed in the past, by piling up a layer appropriately by the approach by the layer. When the version of the components same in two or more piled-up layers is contained, when [ whole ] required, the version of the components which constitute the condition of structure is chosen by choosing the components of a layer piled up later. In addition, "Goldstein, 'Layered Networks as a Tool for Software Development', proceedings of 7 th International Conference on Artificial Intelligence, pp.913-919, and 1981" show the example which used for the version control of software the approach which used the layer.

[0012] Moreover, by "Cellary, 'Consistency of Versions in Object-Oriented Databases', proceedings of International VLDB conference, pp.432-441, and 1990" which described the version control of an object oriented database, by matching a version name with the components belonging to the whole specification structure shows the technique of making restoration of the specified whole version structure easy. moreover "Osterbye which described the version control of a hypertext, and Structural and Cognitive Problems in Providing Version Control for Hypertext' and Proceedings of ACM Conference on hypertext (ECHT'92), By pp 33-42 and 1992.", by choosing a suitable thing from the version of each node contained in a hypertext on condition that the date and time of creation etc. shows the technique of constituting a configuration.

[0013] Problem(s) to be Solved by the Invention] As mentioned above, although it is possible to restore the version of the past document if it is in the conventional version control, and the whole document structure is fixed, in edit of a actual document, the case where a change of the whole document structure is also made with modification of a content block as shown in drawing 86 - drawing 89 is not new. Moreover, in the process of actual document writing, there is a case where the need of restoring the document before edit arises, plentifully like example (1) - (3) shown below to the document with which edit was

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

made.

0014] (1) In order to take out the document written to the ejection, for example, the already deleted past, of the document in specific time, specify time and restore the version of the document which was being edited at the event of the past.

(2) The ejection of the document used for the specific object, for example, an interim report etc., restores the version of the document which packed temporarily and was distributed in the old writing process.

(3) When writing is made, the ejection, for example, two or more persons in charge, of the document updated by the specific man, in order to check the content, restore the version before updating by the specific man is performed, and the version after being carried out.

0015] Moreover, it may be necessary to take out the structure of the document which did not exist in the past other than the need for restoring the version which existed in the past. It is because various demands which it is not beforehand determined in many cases and were not predicted with progress appear how generally work advances. therefore -- even if it opts for the procedure of a person in charge and merge exactly beforehand and advances writing of a document -- on the way -- coming out -- nice -- Yuka -- there are many things. In order to fill a demand, the need of reconfiguring the document made by then according to the situation which was not assumed at the beginning arises. For example, while writing one investigation report by two or more persons, when it is necessary to take in the newest information that other members wrote to \*\*\*\*\* of the report made from another object in the not \*\*\*\*\* but past of the newest report under present writing and a document needs to be made, the document structure which must be taken out is not what existed in the past, and becomes what was compounded from the past components for the new object.

0016] However, if it was in the conventional version control, since it was that whose restoration of the past document is attained on the assumption that the whole document structure is fixed, restoration processing adapted to the above actual document processing system activities was not able to be performed. In addition, even if it is the case where a content block and the whole document structure are changed freely, if the version of both the whole document structure and a content block is specified, it is possible to realize restoration for the past document structure, but in order that for that a user may grasp the whole picture of the modification process of the whole document structure, it will be forced an activity very difficult for a user, and complicated, and is not practical.

0017] Furthermore, it is more desirable to be able to change consistently also about content blocks other than one's range in its duty, since the effect of modification of the content block which he is taking charge of will lead to modification of other content blocks in process of actual document writing in many cases, if it says. That is, in process of actual document writing, it may also be necessary to edit the content block under edit simultaneously by other persons in charge, or to refuse other persons in charge and to change the whole document structure that there is nothing. Thus, when each person in charge person performs edit etc. freely, even if it is, if the document of the past for every each person in charge person can be restored, it will be convenient, but if it is in the conventional version control Since it was that whose restoration of the past document is attained on the assumption that the whole document structure is fixed, each person in charge person always had to protect the constraint on edit, and was doing the writing activity the troublesome thing.

0018] In addition, which node is multiplexed how, and since these nodes are altogether left to the user about of what kind of distinction the method is carried out, a user has forced very complicated actuation, although it is also possible to restore the document of the past for every person in charge by specifying the attribute given to the node if it is in the above-mentioned multiplex document processing system system (JP,7-44563,A, JP,6-131343,A). Furthermore, if it says and will consider using the multiplex document processing system system concerned for co-authoring or restoration of the version in a certain event, it has the capacity to manage two or more documents which two or more persons in charge drew up and with which the contents differ selectively, but in order to enable co-authoring and restoration of a required version actually, an agreement complicated about the attribute given the timing of multiplexing of a node and then is needed. Therefore, very complicated actuation and information were needed on the occasion of actual utilization, and the multiplex document processing system system concerned was not able to be substantially said to be the means for supporting restoration and co-authoring of a version.

0019] Moreover, since the conventional version control technique was a technique which chooses a required thing out of the structure which existed in the past intrinsically, it was not able to take out suitable structure according to the new situation which was not assumed at the beginning according to directions of a user. For example, by the approach using a derivation tree, the structure which can be taken out is in the past condition expressed as a node of a derivation tree, and by the approach of "Cellary", since it names the version which existed in the past, it can take out only the structure which existed in the past. moreover, although the approach by the layer can also make the structure which was not in the past by changing the sequence of a layer, it cannot know whether structure to take out can be taken out and the thing which should first determine the sequence of a layer how in order to take out. Therefore, a user can direct suitable structure for corresponding to a new situation, and cannot take it out. Moreover, although possibility that required structure can be taken

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

but is shown if conditions also with the suitable approach of "Osterbye" are given, it is not shown what kind of conditions should be given actually. Therefore, a user can direct suitable structure for corresponding to a new situation, and cannot make it out.

[0020] Even if this invention makes a user edit freely about the structured data which was made in view of the above-mentioned conventional situation, and makes the graph structure of document data etc., it aims at closing creation of the suitable DS doubled with restoration and the new situation of the past DS, if . Furthermore, speaking concretely, this invention's aiming at realizing restoration and co-authoring of the past document without trouble, even if it makes a writer edit a document freely, without giving the constraint on edit. Moreover, when the whole document structure is changed, even if there is this invention, it aims at constituting the structured statement document which can restore the document version of the required past without trouble. Moreover, this invention aims at restoring the document version of the suitable past by simple assignment actuation by the user. Moreover, this invention is carried out by simple assignment actuation by the user based on the document made in the past, and aims at making the suitable document compared with the new situation which was not assumed at the beginning. [0021]

[Means for Solving the Problem] In the structured data processor of this invention, the node information on structured data have accomplished the graph structure for a node maintenance means holds, and while hold the link information which expresses the related information between the nodes of structured data to a link maintenance means and holding considering the link concerning the same edit processing of two or more links which make the same node the starting point at a links relation maintenance means as a set, the physical-relationship information during the link where it is contained under each set holds. And by specifying any one set under link set held at the link relation maintenance means, the structured data used as the object for edit is specified based on the information held at the node maintenance means and the link maintenance means, is managed with the current structure management management tool, and carries out edit processing by the edit means in this specified structured data. And when modification is added to a link by edit processing, the link set including the link where the registration means was changed is generated, and it is made to hold for a links relation maintenance means.

[0022] In addition, said edit means also performs processing which adds a node, and said registration means also performs processing which makes the added node hold for a node maintenance means. Moreover, it has further a selection attribute matching means to give a selection attribute to the link set changed by edit processing as information concerning edit processing, and when giving version information especially as a selection attribute, a selection attribute matching means gives version information also to other links in the structured data applied to edit processing.

[0023] Moreover, in the structured data processor of this invention, an input of the selection condition for making structured data reconfigure from a condition input means reconfigures structured data based on the link set as which the link information to which the selection attribute corresponding to the selection condition into which the link set selection means was inputted is given was chosen as from the links relation maintenance means, and the data configuration means was chosen. In addition, the link set selection means follows the selection rule [ say / giving priority to version information etc. ] over which priority is given to the time information over which priority is given to user information when two or more link sets corresponding to the inputted selection condition exist and which were set up beforehand, and chooses a gap or one link set, and a data configuration means reconfigures structured data based on the link set concerned.

[0024] Here, when this invention is applied to processing of structured statement document data in which the tree structure is accomplished, this invention is realized as the following document processing system equipments or the document processing system approach. In addition, in the following publications, a logical unit corresponds to the unit in the DS of a node and a link which contains either at least (partial data).

[0025] Make either hold as a logical unit at least, and the group (a set) of the logical unit which consists of all the logical units that have a brotherly relation in the tree structure of the reference unit showing a related location with the structure of the content unit which is a part of content of the structured statement document which has constituted the tree structure for the logical unit maintenance means, or a hierarchy [ directly under ] which constitutes a structured statement document for a logical unit group maintenance means makes hold with the document-processing-system equipment of this invention. In addition, a reference unit expresses a set of at least one logical unit group located directly under this reference unit, and the whole document is constituted by the logical unit group shown per reference. Moreover, the relation of logical units which has the same logical unit (parents) right above is called brotherly relation.

[0026] Furthermore, in this writing processor, when document structure is edited with a document edit means, the logical unit group by whom the logical unit group generation means newly generated the logical unit group corresponding to the document edit concerned, and the means with reference relation was newly generated, and a predetermined reference unit is associated. Consequently, the newly generated logical unit group is included in the existing structured statement in the latter, and the structured statement document according to document edit is drawn up. Moreover, the newly generated

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

logical unit group is added to the same set in the logical unit group before corresponding edit, and is included in a structured statement in the letter, and these logical unit group is collected into a set of the same hierarchy, and is held at a logical unit group maintenance means.

[0027] Moreover, since matching and a criteria specification means constitute document structure for the selection information for choosing a logical unit group and this logical unit group with a selection information matching means from the document processing system equipment of this invention further, if the selection condition containing said selection information is specified, based on the selection information matched by the selection information matching means, the logical unit group by whom the logical unit group selection means is held at the logical unit group maintenance means will be chosen. And a document configuration means constitutes a document based on the logical unit group chosen by the logical unit group selection means.

[0028] As this selection information, the version information of a document, the updating time of a document, the user information on a document, etc. are used, and it is matched with the logical unit group by whom these selection information is held with the selection information matching means at the logical unit group maintenance means, for example. Therefore, according to the version information of the document with which the document constituted based on selection information was specified, the updating time of a document, the user information on a document, etc., the past document version is restored based on these selection information.

[0029] With the document processing system equipment of this invention, further moreover, the selection information matched with a logical unit group by the selection information matching means The selection condition which considers as version information, updating time information, and user information, and is specified by the criteria specification means with the retrieval means with which the logical unit group selection means was equipped including the version, time information, and user information on a document Henceforth [ the updating time information matched with the logical unit group who is in agreement with the version information specified by the criteria specification means ] Search the logical unit group who is matched with the updating time information before the specified time information, and is in agreement with the specified user information, and further with the selection means with which the logical unit group selection means was equipped It is as a result of [ by the retrieval means ] retrieval, and when two or more logical unit groups exist, any one logical unit group is chosen. And a display means displays this selected logical unit group, and the document of the newest version on the basis of the time which corresponded to the user and was specified as him is searched and displayed.

[0030] Moreover, if the object for edit specifies out of the reference unit which expresses with the document-processing-system approach of this invention the structure of the content unit which is a part of content of the structured statement document, or a low-order hierarchy and this content unit or reference unit that specified edits, the logical unit group who consists of all the content units or reference units which have a brotherly relation in the tree structure which constitutes a structured statement document will be generated corresponding to edit. And if the selection information for choosing the newly generated logical unit group is inputted, the inputted selection information and the newly generated logical unit group will be associated, and this logical unit group will be added to a set of a logical unit group including said specified content unit or reference unit. Therefore, the newly generated logical unit group is included in the existing structured statement in the letter, and the structured statement document according to document edit is drawn up.

[0031] Moreover, if the selection condition for choosing the structured statement document which has the reference unit which expresses with the document processing system approach of this invention the structure of the content unit which is a part of content of the document, and a low order hierarchy is inputted Out of the logical unit group who consists of all the content units or reference units which have a brotherly relation in the tree structure which constitutes a structured statement document A logical unit group is chosen based on the selection information matched with the inputted selection condition and the logical unit group, and a document is constituted and displayed from the selected logical unit group. Therefore, the document corresponding to the selection information concerned is constituted and displayed by inputting the selection condition containing selection information, such as version information, updating time information, and user information.

[0032] That is, according to this invention, even if it makes possible version control applicable fair about the whole document structure and substructure and adds modification to the content and structure of a document freely, it is possible to take out the document of a required version. If it puts in another way, the whole document consists of this inventions as what contained all the versions of the component of a document as document data combining the substructure extracted according to the suitable extraction condition out of this as DS which can manage the version of whole structure and a substructure for the structure of a document equally.

[0033] Moreover, according to this invention, in co-authoring, it becomes possible to perform edit processing, without editing consistently the document's of the version which suits the conditions' which each person in charge gave making it possible, and each person in charge person getting confused. In this invention, namely, version assignment information and user assignment information, By defining the conditions for taking out the version which a common writer should make the object

**BEST AVAILABLE COPY**





of edit combining time assignment information, and relating such information with the target document part in the case of each person in charge person's writing activity Even if it edits a document freely, the document of a version which should be edited for each writer is restored by constituting a document based on these information.

[0034]

[Embodiment of the Invention] In explaining the document processing system equipment and the approach concerning one example of this invention, the configuration and version control of a structured statement document in this example are first explained with reference to drawing 3 - drawing 8 . In this example, the content unit which was indicated to the claim and which is a part of structured statement document which has constituted the tree structure in addition, as a content block The reference unit which shows the related location for the lower layer structured division of a structured statement document as support It realizes, and further, either is made into a logical unit at least, and selection information for the logical unit group who consists of the logical unit of these content blocks and support located in the same hierarchy of the tree structure to choose a logical unit group as a stream is realized as a stream attribute.

[0035] In this example, it makes it possible to manage the version of the whole document structure and a content block with an equivalent method, as the structured statement document of the tree structure based on a stream is used and the whole document structure and substructure can be systematically treated by managing a version per stream. That is, in this example, generation of the document structure of arbitration, modification, restoration, etc. are attained by combining the basic operation concerning edit of general document structures, such as modification of the content of a document, insertion of the content of a document, or the substructure of a document, deletion of the content of a document, or the substructure of a document, modification of the sequence of the content of a document, or the substructure of a document, and the content of a document, migration of the substructure of a document, and actuation of generation of a suitable content block or a stream.

[0036] In the example of the document structure shown in drawing 3 , when Node A is used as the support of the document root, each stream c and d in which the stream b in which the stream a containing Support B and a content block (u) contains 1 chapter and Support C and D, for example contains a knot, a content block (v), and (x) will correspond to a term. And in performing edit processing in which create a new content block and it changes for the existing content block, processing in a stream unit as shown in drawing 4 - drawing 8 is performed, and it changes the content block and whole structure of a document.

[0037] First, in changing a content block to the original document shown in drawing 3 , as shown in drawing 4 , the stream l2 containing a new content block (y) is made, and while replacing the stream d1 containing the existing content block (x), and the new stream d2, it carries out a storage management as a set to which Support D refers to the old stream d1 and the new stream d2.

[0038] moreover, in inserting a new substructure to the original document shown in drawing 3 As shown in drawing 5 , while making the new stream b2 which contains Support E further to the existing stream b1 containing Support C and D While making the new stream e containing a content block (z) and replacing the new stream b2 with the existing stream b1, each support C, D, and E of the new stream b2 is made to refer Streams c, d, and e, respectively. In addition, even if it is in this case, the storage management of the same hierarchy's streams b1 and b2 referred to of the same support B is carried out as a set which Support B refers to.

[0039] Moreover, while making the new stream b3 which deleted Support D to the existing stream b1 containing Support C and D and replacing the existing stream b1 and the new stream d3 as shown in drawing 6 in deleting the existing substructure D to the original document shown in drawing 3 , a storage management is carried out as a set whose support B refers to the old stream b1 and the new stream b3. Moreover, while making the support C of the new stream b3 refer Stream , the storage management of the stream d is carried out as a stream referred to with the support D of the old stream b1.

[0040] Moreover, while making the new stream b4 which replaced the sequence of Support C and D to the existing stream 1 containing Support C and D and replacing the existing stream b1 and the new stream d4 as shown in drawing 7 in replacing the sequence of Substructures C and D to the original document shown in drawing 3 , a storage management is carried out as a set whose support B refers to the old stream b1 and the new stream b4.

[0041] moreover, in moving Substructure D to the original document shown in drawing 3 As shown in drawing 8 , while making the new stream b3 which deleted Support D to the existing stream b1 containing Support C and D and replacing the existing stream b1 and the new stream d3 The new stream a2 in which Support D was further included to the existing stream a1 containing a content block (u) and Support B is made, the existing stream a1 and the new stream a2 are replaced, and Stream d is made to refer to as it is to Support D. In addition, even if it is in this case, the storage management of all the streams a1 and a2 referred to of the same support A is carried out as a set which Support A refers to, and they carry out the storage management of all the streams b1 and b3 referred to of the same support B as a set which support B refers to.

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

0042] As mentioned above, the stream by which edit processing was carried out is memorized as the same set as the stream before edit, and these streams by which grouping was carried out are referred by the same support. Therefore, when the whole document configuration is changed by edit processing by choosing the stream which suits [ from ] predetermined conditions among the sets shown of support, and constituting a document by these streams so that it may mention later, even if it is, the document of the past before edit can be restored.

0043] The document processing system equipment of this example which performs such a document processing system as the configuration shown in drawing 1 . The logical unit maintenance means 2, the logical unit group maintenance means 3, and the selection information matching means 4 are constituted by the memory 1 which consisted of semiconductor memory, magnetic storage, etc., and the storage management of the information about a structured statement document is carried out to it with these means 2-4. The logical unit maintenance means 2 holds the logical unit of support or a content block which contains either at least, and the logical unit group maintenance means 3 holds the stream which consists of a logical unit.

0044] It is possible to choose the stream which corresponds from the logical unit group maintenance means 3 by the selection information matching means 4 matching the stream attribute for choosing the stream and stream which are held at the logical unit group maintenance means 3, and specifying a stream attribute by this. Therefore, the structured statement document is stored in memory 1 as DS which matched the stream attribute for every stream, as shown in drawing 2 , and it defines the group of these streams and a stream attribute by explanation of the following in this example as a stream set. Moreover, two or more groups of the stream which a stream set is referred by the same support and located in the same hierarchy in document structure, and a stream attribute may be included, and a stream set serves as a set which included two or more streams in this case. Moreover, the top node of document structure is root support, and as further said to the low-ranking stream set from the support contained in the stream set concerned from this root support to the low-ranking stream set, these stream set is tree-structure-ized by reference by support.

0045] The document edit means 5 performs edit processing to the structured statement document currently held at memory based on directions by the user from input means, such as a keyboard and a mouse, and performs modification processing to a content block and a substructure (stream) as shown in drawing 4 - drawing 8 . The logical unit group generation means 6 generates the new stream d2 containing a content block (y), when a content block (x) is changed into a content block (y), as processing which generates the new stream which corresponds based on edit processing having been made with the document edit means 5 is performed, for example, it was shown in drawing 4 .

0046] As the reference correlation means 7 performs processing which adds the newly generated stream to the same set as the stream of the origin used as the object for edit, and is made to hold for the logical unit group maintenance means 3 with the logical unit group generation means 6, for example, showed it to drawing 4 , when a new stream d2 is generated to the existing stream d1, the stream d2 concerned adds to the same set (a stream set) as a stream d1.

0047] The criteria specification means 8 has input means, such as a keyboard which a user operates, and a mouse, and if a selection condition including a stream attribute is inputted, it will output this to the logical unit group selection means 9. In addition, in this example, the version information of a document, the updating time information on a document, the user information on a document, etc. are set up as a stream attribute, and a selection condition including such a stream attribute is inputted into the logical unit group selection means 9. The logical unit group selection means 9 has the retrieval means 10 and the selection means 11, in order to perform processing which suits a selection condition from the logical unit group maintenance means 3 based on the selection condition inputted from the criteria specification means 8 and which makes stream selection and to perform this selection processing.

0048] The retrieval means 10 carries out retrieval processing of whether the stream which suits a selection condition exists in the stream currently held at the logical unit group maintenance means 3, and when the stream which the selection means 11 suits by this retrieval exists, it performs processing which chooses the stream concerned. For example, when the version, time information, and user information on a document are specified as a selection condition, the stream which is in agreement with the user information which was matched with the updating time information before the specified time information, and was specified by the retrieval means 10 henceforth [ the updating time information corresponding to version information ] is searched, and the stream which suits this selection condition with the selection means 11 is chosen further. In addition, the selection means 11 chooses from the stream which is in agreement with version information among this attribute, when the stream which suits all the attributes included in a selection condition as mentioned above by retrieval by the retrieval means 10 does not exist.

0049] The document configuration means 12 performs processing which constitutes a document based on the stream chosen by the logical unit group selection means 9, and the document structure display means 13 performs processing which displays on a display 14 the document constituted by the document configuration means 12. The non-displayed information-display means 15 performs processing displayed on a display 14 with the document structure display means 13

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

by making into a marker the stream which is not displayed [ which is not displayed as a component of a document ]. namely, -- the case where the stream set containing the stream chosen by the logical unit group selection means 9 contains other streams -- being concerned -- others -- it considers as the candidate who marker-ized so that a stream may be mentioned later ( drawing 17 ), and it displays on a display 14 with the document constituted by the document configuration means 12.

0050] The logical unit group designation means 16 has input means, such as a keyboard and a mouse, and performs processing which specifies the marker displayed by the non-table information-display means 15 based on the input from a user. The logical unit group reselection means 17 makes the processing which changes the stream corresponding to the specified marker concerned, and the stream already chosen with the logical unit group selection means 9 to the logical unit group selection means 9 perform based on having specified the marker with the logical unit group designation means 16. That is, when the content block (text) of a certain stream is displayed on the display 14 as a component of a document, by choosing a marker, the content block (text) of other streams changes and is displayed. Moreover, when the configurations which relate to support called a number, a location, etc. of support between the replaced streams differ, the whole document configuration currently displayed is also changed.

0051] Here, the selection processing of a stream based on the selection condition which the above-mentioned logical unit group selection means 9 performs is explained in more detail with reference to drawing 9 . In addition, in the example shown in drawing 9 , the user information which consists of user names, such as "Usui", "Sato", and "Tani", as an attribute of each stream, updating time information, and version information are matched, and the criteria time (criteria time information) for processing with reference to a document, the version information (criteria version information) used as criteria, user information, and \*\* are contained in the selection condition inputted. Moreover, in accordance with the time of an axis of ordinate, stream [ for every user information ] (1) - (5) is shown.

0052] In this example, when Mr. "Usui" of a user shows by the dotted line all over drawing as reference time, the case where a document is referred to actually is considered. First, when specifying nothing as a selection condition, the newest stream (5) is chosen on the basis of reference time. Moreover, when only the user information made into the principal (namely, "Usui") is specified, the newest stream (1) whose stream attribute is "Usui" is chosen. Moreover, when only the user information made into "Sato" is specified, the newest stream (5) whose stream attribute is "Sato" is chosen. Moreover, when only the criteria time shown all over drawing is specified, the stream (4) of the newest [ time / of a stream attribute / updating ] to criteria time is chosen.

0053] Moreover, when the time illustrated as criteria time is specified and "Sato" is specified as user information, a stream attribute is "Sato" and the stream (3) of the newest [ time / updating ] even in criteria time is chosen. Moreover, when only the illustrated criteria version is specified, the newest version (5) of a criteria version or later is chosen. Moreover, when the time illustrated as the illustrated criteria version and criteria time is specified, the stream (4) newest even in criteria time is chosen henceforth [ a criteria version ].

0054] Moreover, when the "beginning" is specified as the illustrated criteria version and criteria time, the first stream (2) is chosen henceforth [ a criteria version ]. Moreover, since there is no stream whose stream attribute is "Usui" after a criteria version when a "principal" is specified as the illustrated criteria version as user information, the stream (2) of a criteria version is chosen. Thus, based on a stream attribute, the stream which suits the selection condition specified from the criteria specification means 8 is chosen, and a document is constituted by the document configuration means 12 based on the stream concerned.

0055] Next, it explains using the example which shows the outline of the document edit processing carried out by the document processing system equipment of the above-mentioned configuration to drawing 10 - drawing 36 . In addition, the function by each above-mentioned configuration means will also be further clarified by the following explanation. By this example, Mr. Sato and Mr. Tani are writing the document of the identifier of an "invention description" together by the edit course shown in drawing 10 . In addition, from initiation of each editing task to termination is called a session, initiation is performed by open Lycium chinense in a document with a document edit tool, and closing is performed by closing a document.

0056] At drawing 10 , time amount is taken along an axis of abscissa, and each session of 1-7 is shown by the horizontal segment corresponding to person-in-charge "Sato" and "Tani". Moreover, it is shown that the triangle with the black back end of each segment shows the event of preservation of a document being performed, and defined the identifier of a version by the line of the perpendicular direction from the mark of preservation. For example, by the session 4, a document is saved in January 4, 1995 and defining the version of the identifier of "an interim report" in this case is shown. In addition, by this example, although he is trying, as for each session, for time amount not to lap, this is for not complicating explanation unnecessarily, and this invention does not produce any trouble, even if the time amount of a session laps.

0057] The session 1 when the user Sato draws up and saves a document "an invention description" newly is explained with reference to drawing 11 - drawing 15 . First, as shown in drawing 11 , if Mr. Sato starts the document editor of the document

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

edit means 5 to 15:00 on January 1, 1995, the dialogue tool shown in drawing will be displayed by the display 14, and the input of a Norifumi Arata title will be demanded from a user. On the other hand, Mr. Sato inputs an "invention description" as a document route name, and if the "ok" carbon button which directs the completion of setting out is clicked, the identifier of the root support which starts the extract of a document will be set up. In addition, refer to the stream set which constitutes a document for this root support.

0058] Subsequently, as shown in drawing 12, the document structure which the document editor extracted is displayed on a display 14. The display of this document structure is performed by displaying the marker of a white triangle who shows the location of support, a support name, and the content block of the stream which support refers to. In addition, the depth of the indent of the marker who shows the location of support shows the depth of the hierarchy of the stream in document structure, and the location where a marker is displayed serves as the right in drawing, so that it becomes the support which refers to a lower layer stream.

0059] That is, when the content block (the content of a text) is not yet contained in the stream referred to of the root support of the name an "invention description", as shown in drawing 12 (1), a root support name "an invention description" is displayed and the stream of the empty which the root support concerned refers to is displayed all over drawing as a field shown by the dotted line. And Mr. Sato performs an editing task, and if a text is described as a content block (u) to the stream which root support refers to, as this content of a text shows drawing 12 (2), it will be displayed in the above-mentioned field. As a result of this edit, as the logical unit group generation means 6 shows to drawing 13, the stream a which consists of a content block (u) created newly is generated, and the document structure where this stream a was referred by the root support A is constituted.

0060] Subsequently, edit processing will be ended, the "save" carbon button of the document editor which Mr. Sato shows to drawing 12 is clicked, if the directions which save a document are inputted into 20:00 on January 1, 1995, the dialogue tool shown in a display 14 at drawing 14 will be displayed on it, and the assignment at the time of preservation will be urged to it. On the other hand, in this example, Mr. Sato directs preservation processing of the document which clicked and created the "ok" carbon button, without setting anything up. In addition, the root support name "an invention description" is displayed on this dialogue tool as the document root.

0061] The document structure shown in drawing 15 is stored in memory 1 as a result of this preservation processing. That is, the content block (u) contained in the stream a1 contained in the stream set a referred to of the root support A and the root support A and a stream set and a stream a1 is newly generated by this document structure, and 1995. "1.1 20:00" which is "Sato" which is the information of the user who edited, and the time which saved data (updating) is matched with a stream a1 with the selection information matching means 4 as a stream attribute.

0062] Next, in the session 2 shown in drawing 10, he adds edit to the document "an invention description" drawn up in the session 1, and Mr. Sato saves an edit result in memory 1. First, as shown in drawing 16 (1), the document folder is displayed on the display 14, and if Mr. Sato double-clicks and chooses the "invention description" icon of a document folder to 15:00 on January 2, 1995, the document editor of the document edit means 5 will be started. Consequently, the extraction condition setting-out tool shown in a display 14 at drawing 16 (2) is displayed, and setting out of an extraction condition is urged. On the other hand, variable "% principal" to which Mr. Sato expresses a principal as user information from the criteria specification means 8 is inputted, and edit processing will be started if the "ok" carbon button which directs the completion of setting out is clicked. In addition, "%" notations, such as "% principal" and "% beginning", show that a system is the variable which calculates and sets up a value, when starting a document extract.

0063] As the above-mentioned setting out extracting the document containing the newest component (stream) which the principal updated, and making it applicable to edit is shown and it is shown in drawing 17 (1), a document editor extracts the document structure when saving in a session 1 from memory 1, and displays a document editor on a display 14. Subsequently, it is displayed in the field which support "a claim" refers to as the content of this text shows drawing 17 (2), when a text is described as a content block (v) to the stream which adds a "claim" to the stream which the "invention description" which is root support as Mr. Sato shows drawing 17 (1) as an editing task of a document refers to as support, and support "a claim" refers to further.

0064] As shown in drawing 18, while the new stream a2 which is referred to by the logical unit group generation means 6 with the root support A, and contains a content block (u) and Support B with it is generated as a result of this edit, the new stream b1 which is referred by Support B and contains a content block (v) is generated. In addition, as shown in drawing 17, to a document editor, the candidate marker Ma is displayed by the non-displayed information-display means 15, and it is shown in the stream set which Support A refers to that other streams other than the stream by which it is indicated by the display exist as a candidate. In this case, the candidate of the stream which Support A refers to has a1 and a2, and when indicated by the display about one stream, a display indication of the stream of another side is given by Marker Ma.

0065] Subsequently, edit processing will be ended, the "save" carbon button of the document editor which Mr. Sato shows

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)



to drawing 17 is clicked, if the directions which save a document are inputted into 20:00 on January 2, 1995, the same dialogue tool will be displayed to have been shown in the display 14 at drawing 14 on it, and the assignment at the time of preservation will be urged to it. On the other hand, when preservation processing of a document is directed like the time of Mr. Sato being a session 1, the document structure shown in drawing 19 is stored in memory 1.

[0066] Namely, the stream set a referred to of the root support A and the root support A at this document structure The streams a1 and a2 contained in the stream set a, the stream set b referred to with Support B Stream b1\*\* contained in the stream set b is contained. To a stream a1, further "Sato" of user information, The stream attribute of 1995. "1.1 20:00" of data storage (updating) time information, The stream attribute of 1995. "1.2 20:00" of "Sato" of user information, and data storage (updating) time information in a stream a2, The stream attribute \*\* 1995. "1.2 20:00" of "Sato" of user information and data storage (updating) time information is matched with a stream b1 by the selection information matching means 4.

[0067] Next, in the session 3 shown in drawing 10, Mr. Sato adds edit to the document "an invention description" which he edited in the session 2 further, and he saves an edit result in memory 1. First, if Mr. Sato starts edit of a document "an invention description" like a session 2 to 15:00 on January 3, 1995, a document editor will be displayed on a display 14, as he document structure when saving in a session 2 is extracted from memory 1 and it is shown in drawing 20 (1).

[0068] On the other hand, if Mr. Sato adds an item 2 to the column of a claim as an editing task of a document as shown in drawing 20 (1), the logical unit group generation means 6 will create the stream containing the support C of an item 1, and the support D of an item 2 as a stream which the support B of a claim refers to. And if a text is described in this item 2 as shown in drawing 20 (2), the content block (x) which consists of this text will be generated by the logical unit group generation means 6 as a new stream referred to with Support D.

[0069] Therefore, the stream a2 which is referred by the root support A and contains a content block (u) and Support B as shown in a document editor at drawing 21, The stream b2 which is referred by Support B and contains Support C and D, the stream c1 which is referred by Support C and contains a content block (v), the stream d1 which is referred by Support D and contains a content block (x), and \*\* are generated. In addition, since the stream which Support B refers to by the above-mentioned edit processing serves as two or more candidates, as shown in drawing 20, a display indication of the marker Mb who shows the stream which is not displayed of these streams b1 and b2 is given.

[0070] Subsequently, edit processing is ended, and when the directions whose Mr. Sato saves a document like a session 2 20:00 on January 3, 1995 are inputted, the document structure shown in drawing 22 is stored in memory 1. Namely, the stream b2 which contains Support C and D in the stream set b is added to this document structure to the structure shown in drawing 19. Furthermore, stream d1\*\* with the content block (x) contained in the stream set d referred to of the stream c1 with the content block (v) contained in the stream set c referred to of Support C and the stream set c and Support D and the stream set d is added. Furthermore, to the newly [ this document structure ] added streams b2, c1, and d1, the stream attribute of 1995. "1.320:00" of "Sato" of user information and data storage (updating) time information is matched by the selection information matching means 4, respectively.

[0071] Next, in the session 4 shown in drawing 10, Mr. Tani adds edit to the document "an invention description" which Mr. Sato drew up, and the edit result is saved as an "interim report" version. First, if Mr. Tani starts edit like a session 2 to 10:00 on January 4, 1995, as shown in drawing 23, an extraction condition setting-out tool will be displayed on a display 14. On the other hand, Mr. Tani clicks the "ok" carbon button, without specifying anything as setups, he indicates the document editor as shown in drawing 24 by the display, and performs edit. In addition, if edit is started without setting up any conditions in this way, it is not concerned with a user, but the newest document will be extracted from memory 1 and will serve as an object for edit. Moreover, since Mr. Tani is not updating the document, if "% principal" is directed from the criteria specification means 8 as user conditions like Mr. Sato, he will be extracted at all as a document here until now.

[0072] As shown in drawing 24, a document editor extracts the document structure when saving in a session 3, and is indicating by the display, and an item 1.5 is added between an item 1 and an item 2 as a stream which the support B of a claim refers to as Mr. Tani shows in drawing as an editing task. As a result of this edit, as shown in drawing 25, the new stream b3 which is referred to by the logical unit group generation means 6 with Support B, and contains Support C, D, and E with it, and the new stream e1 containing the content block (z) of the item 1.5 referred to with Support E are generated.

[0073] Subsequently, edit processing will be ended, Mr. Tani clicks the "save" carbon button of a document editor, if the directions which save a document are inputted into 15:00 on January 4, 1995, the dialogue tool shown in a display 14 at drawing 26 will be displayed on it, and the assignment at the time of preservation will be urged to it. On the other hand, Mr. Tani sets up "an interim report" as version information, and he directs preservation processing of the document which he clicked and created the "ok" carbon button.

[0074] The document structure shown in drawing 27 is stored in memory 1 as a result of this preservation processing. That is, the new stream b3 containing Support C, D, and E is added to the stream set b referred to of Support B to the document structure in the session 3 shown in drawing 22, and new stream e1\*\* further contained in the stream set e referred to with

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

Support E and the stream set e is added. In addition, "an interim report" which is "1995. 1.4 15:00" and version information which are "Tani" which is the information of the user who edited as a stream attribute, and the time which saved data (updating) is matched with the streams b3 and e1 generated by this edit processing by the selection information matching means 4. Moreover, by the selection information matching means 4, "an interim report" of version information adds also to the stream attribute of the streams a2, c1, and d1 concerned with this edit processing, and is matched with it.

[0075] Next, in the session 5 shown in drawing 10, Mr. Sato adds edit to the document "an invention description" which he edited further, and he saves an edit result. In addition, although modification by Mr. Tani is added in the session 4 at the time of this edit initiation, edit is resumed about the document in the condition that he edited and saved by setting out of an extraction condition. That is, document editor starting is carried out like a session 2 15:00 on January 4, 1995. And by making into "% principal" assignment of the extraction condition setting-out tool inputted from the criteria specification means 3 in this case, the logical unit group selection means 9 chooses not b3 that Mr. Tani created but b2 which Mr. Sato created as a stream which the support B of a claim refers to, and as shown in drawing 28 (1), it expresses the document structure in the condition that Mr. Sato saved in the session 3 as a document editor.

[0076] In addition, the candidate marker Mb in this case shows that there is a new stream by the candidate of the stream in which inverse video is carried out and the support B of a claim is carrying out current reference by this. And as shown in drawing 28 (2), when Mr. Sato rewrites the content block (x) of an item 2 to a content block (y) as an editing task, as he shows drawing 29, the logical unit group generation means 6 newly generates the stream d2 which rewrote the content block (x) to the content block (y) as a stream which the support D of item 2 \*\* refers to.

[0077] Subsequently, when Mr. Sato saves a document like a session 1 20:00 on January 4, 1995, the document structure shown in drawing 30 is stored in memory 1. That is, the new stream d2 containing a content block (y) is added to the stream set d referred to of Support D to the document structure in the session 4 shown in drawing 27, and "1.4 20:1995. 00" \*\* which are "Sato" which is the information of the user who edited as a stream attribute, and the time which saved data (updating) is further matched with this stream d2 by the selection information matching means 4.

[0078] Next, in the session 6 shown in drawing 10, the result of the edit added by writers other than itself is referred to after that with the "interim report" version of the document "an invention description" which he added edit and Mr. Tani saved. A document editor is started in Mr. Tani starting edit of a document "an invention description" to 10:00 on January 5, 1995, and choosing and double-clicking the icon of a document folder like a session 2 to it. At this time, with the extraction condition setting-out tool shown in drawing 31 (1), it is urged to setting out of the extraction condition of a document, on the other hand Mr. Tani specifies "% principal" as "interim report" and user information as version information. The document which changes from the component (stream) updated by Mr. Tani to the version specified as "an interim report" by this assignment is extracted from memory 1, and serves as an object for edit.

[0079] That is, the stream to which the logical unit group selection means 9 includes "an interim report" in an attribute according to the conditions inputted from the criteria specification means 8 is searched, and the newest stream which Mr. Tani created is chosen. Here, the stream d1 in the event of Mr. Tani setting up not the newest stream d2 to which Mr. Sato created him as a stream which the support D of an item 2 refers to since Mr. Tani was not editing but "an interim report", although Mr. Sato was editing after setting up a version in the session 4 is chosen. Consequently, as shown in drawing 32 (1), the document structure (drawing 24) of "an interim report" which Mr. Tani saved in the session 4 is displayed on a document editor. In addition, it is shown that the inverse video of the candidate marker Md corresponding to the support D of an item 2 is carried out, he is created by Mr. Sato, and depends, and there is a candidate of a new stream.

[0080] Here, if an extraction condition setting-out tool can be called again and conditions are changed with this called tool by clicking the "filter" carbon button of a document editor, document structure will be re-extracted on new conditions. Therefore, assignment except "% principal" is performed as an extraction condition as Mr. Tani shows drawing 31 (2), a user will be fair and a newer stream will be chosen by this assignment. Consequently, the newest stream d2 which Mr. Sato created is chosen as a stream which the support D of an item 2 refers to, and the document shown in drawing 32 (2) is displayed on a document editor.

[0081] Therefore, in assignment of the first conditions shown in drawing 31 (1), in an editor, the document structure shown in drawing 33 (1) is taken out, and the document structure shown in drawing 33 (2) is taken out after modification of the conditions shown in drawing 31 (2). That is, although the stream d1 in which Support D contains a content block (x) is referred to with the document structure shown in drawing 33 (1), the stream d2 in which Support D contains a content block (y) is referred to with the document structure shown in drawing 33 (2). In addition, although Mr. Tani ends edit in a session 6, the document structure which does not save a document but is stored in memory 1 is still the structure shown in drawing 30.

[0082] Next, in the session 7 shown in drawing 10, the thing in which Mr. Sato made the result of edit by writers other than himself reflect about the document "an invention description" into which he has edited is saved as a "last report" version. In

**BEST AVAILABLE COPY**

Sheet  
PAGE BLANK (USPTO)

a session 7, "Mr. Sato makes 15:00 on January 5, 1995 start a document editor like a session 2, and edit of a document "an nvention description" is started.

0083] As shown in drawing 34 , in a session 5, a document editor extracts document structure when Mr. Sato saves from memory 1, and indicates by the display. This document editor is shown that a newer non-chosen candidate is in the stream which the support B of a claim refers to by the candidate marker Mb. If this candidate marker Mb is clicked, a candidate's streams b1, b2, and b3 will be displayed as a menu K, if a stream is chosen from this menu K with the logical unit group designation means 16, the stream chosen by the logical unit group reselection means 17 will be replaced with the stream under current reference, and low-ranking structure will be further reconfigured according to an extraction condition from the replaced stream.

0084] Therefore, when Mr. Sato chooses the newest stream b3 from the candidate menu K, the same document as having created in the session 5 is made for edit. In edit processing of this session 7, the document structure in the session 5 shown in drawing 29 is changed into the structure shown in drawing 33 (2) within a document editor. And if Mr. Sato performs the directions which save a document to 20:00 on January 5, 1995, a display indication of the dialogue tool shown in drawing 35 will be given, and the conditioning at the time of preservation will be urged to it. On the other hand, when Mr. Sato saves by specifying "the last report" as version information, the document structure where edit processing was added as shown in drawing 36 is stored in memory 1. That is, to the document structure in the session 5 shown in drawing 30 , by the selection information matching means 4, the "last report" of version information adds to the attribute of the streams a2, b3, c1, d2, and e1 which were being referred to at the time of preservation processing, and is matched with it.

0085] Next, with reference to the flow chart which shows the processing which extracts a document from memory 1 and is displayed on a display 14 to drawing 37 - drawing 40 , it explains in more detail. As shown in drawing 37 , processing which the logical unit group selection means 9 chooses root support from memory 1 according to the directions from the criteria specification means 8 by the user (step S1), extracts first the structure where it is located in the hierarchy of the low order of the root support specified succeedingly from memory 1, and displays a document on a display 14 is performed (step S2).

0086] Low order structure extract display processing in this step S2 is made in the procedure shown in drawing 38 . First, root support is included in the document structure which the document configuration means 12 is extracting (step S11), and the document editor with which the document structure display means 13 displayed this root support name is displayed on the screen of a display 14 (step S12). Subsequently, the logical unit group selection means 9 chooses from memory 1 the stream set which root support is referring to (step S13), and the stream of the attribute which suits the conditions directed by the user from the criteria specification means 8 is chosen from this stream set (step S14).

0087] Subsequently, it includes in document structure while the document configuration means 12 is extracting the selected stream (step S15), and the document structure display means 13 displays this stream (content block) into the document editor currently displayed on the screen of a display 14 (step S16). Subsequently, the stream as which the non-displayed information-display means 15 was chosen judges whether it is the only stream in the above-mentioned stream set (step S17), and processing is ended when other streams cannot be found into a stream set. In addition, when the support contained in a stream is referring to a low-ranking hierarchy's stream set, the same selection display processing as the above is repeatedly performed about these stream set and a stream.

0088] On the other hand, when other streams exist in [ other than the selected stream ] a stream set It judges whether a stream is a stream newer than the selected stream from the stream attribute (step S18). the non-displayed information-display means 15 -- being concerned -- others -- In being a newer stream, while carrying out the inverse video of the candidate mark showing the stream concerned into the document editor on a display 14 (step S19) In not being a new stream, into the document editor on a display 14, it is noninverting and displays the candidate mark showing the stream concerned (step S20).

0089] Here, if the above-mentioned stream display processing (step S16) is explained in more detail, the processing concerned will be performed in the procedure shown in drawing 39 . First, it judges whether the selected stream is an empty stream (step S31), and in being the stream of the empty which does not include logical units, such as a content block or support, it ends processing as it is. On the other hand, when the logical unit is included in the stream, one unsettled thing in these logical units is chosen (step S32), and it judges whether the selected logical unit concerned is support (step S33).

0090] As a result of this decision, in being support, in [ which displays low order structure which performs low order structure extract display processing shown in drawing 38 and is referred to of support ] being a content block, on the other and (step S34), it carries out display processing of the content block concerned (step S35). And when repeating and performing processing after step S32 when judging whether the above-mentioned processing was completed (step S36) and having ended about no logical units in the selected stream, and having ended, the stream display processing concerned is ended.

0091] Moreover, if the above-mentioned stream selection processing (step S14) is explained in more detail, the processing

TEXT AVAILABLE COPY

10/13/2013  
FREE BLANK (USPTO)

concerned will be performed in the procedure shown in drawing 40 . First, all the streams under stream set are made a candidate list (step S41), and it judges whether the conditions inputted from the criteria specification means 8 have assignment of version information (step S42). When there is version assignment as a result of this decision, a stream with the attribute which suits conditions from SUTORIMU under candidate list is looked for, and this stream is once set up as a criteria stream (step S43).

0092] Subsequently, when it judges whether the conditions inputted from the criteria specification means 8 have assignment of criteria time information (step S44) and there is time assignment, the stream by which time newer than the specified time is set as the attribute is deleted from a candidate list (step S45). That is, the stream by which edit processing was carried out after the time which the user makes the processing object is excepted. Subsequently, when it judges whether the conditions inputted from the criteria specification means 8 have assignment of user information (step S46) and there is assignment of a user, the stream by which the specified user information is not set as the attribute is deleted from a candidate list (step S47). That is, it excepts except the stream concerning the self edit which the user makes the processing object.

0093] Subsequently, whether the criteria stream is chosen judges (step S48), and when chosen, a stream older than a criteria stream is deleted from a candidate list (step S49), and it checks that the candidate list does not serve as empty (step S50), and chooses as a stream which suits the conditions into which the criteria stream was inputted from the criteria specification means 8 (step S54). When judging whether the candidate list is empty on the other hand when the criteria stream does not remain during a candidate list as a result of the above-mentioned decision (step S52) and having become empty, in order to show that which stream is not chosen, it considers as the stream which had the empty stream chosen (step S53).

0094] Moreover, in the above-mentioned decision, a criteria stream is not chosen, but when a criteria stream is not empty (step S52), or when [ even if a candidate list deletes a stream older than a criteria stream from a candidate list, ] it is not empty (step S50), let it be the stream which had the newest stream under candidate list chosen (step S54). Therefore, based on a stream attribute, the stream which suits the conditions inputted from the criteria specification means 8 is chosen.

0095] Next, by choosing the candidate markers Ma and Mb and ... as which the user was displayed into the document editor explains in more detail with reference to the flow chart which shows the processing which extracts the document which replaced the stream from memory 1, and is displayed on a display 14 to drawing 41 . First, if a user directs a candidate marker with the logical unit group directions means 16, it will choose from the document structure in which the stream set with which the stream the non-displayed information-display means 15 was instructed to be belongs is stored by memory 1 (step S61), and will display into a document editor by using each stream under selected stream set as the candidate menu K (step S62).

0096] And a user directs with the logical unit group directions means 16. When it judges whether the stream other than the stream by which a current selection indication of [ in the candidate menu K ] is given was chosen (step S63) and another stream is chosen It changes for the stream which the logical unit group reselection means 17 canceled the structure of the flow order hierarchy of the stream currently displayed by the current document editor (step S64), and canceled further the stream which it reselected (step S65). And the document configuration means 12 constitutes the document which consists of this replaced stream, the document concerned is redisplayed in a document editor, and the display means 13 ends processing (step S66).

0097] Next, with reference to the flow chart which shows the processing which saves the document which carried out edit processing in memory 1 to drawing 42 , it explains in more detail. First, it judges whether it is contained in the root support set of [ in the document structure where the root support of document structure edited by the document editor is stored in memory 1 by the document edit means 5 ] (step S71). Consequently, since the edited document is a completely new document when there is nothing on a root support list, the means 7 with reference relation adds the root support of the document structure concerned to a root support list (step S72).

0098] Subsequently, the means 7 with reference relation relates with a predetermined stream the content block generated by the document editor (step S73), the generated streams are collected (step S74), and the following processings choose the stream which is not yet made (step S75). Subsequently, when the stream set with which an unsettled stream corresponds is looked for out of the document structure in which it is stored by memory 1 (step S76) and there is a corresponding stream set, an unsettled stream is added to (step S77) and the stream set concerned (step S79). On the other hand, when there is no corresponding stream set, a new stream set is added in document structure (step S78), and an unsettled stream is added to the stream set concerned (step S79).

0099] And the selection information matching means 4 sets up attributes, such as time and a user, to the stream added to the stream set (step S80). A series of above-mentioned processings are repeatedly performed about all the generated

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)



streams (step S81), and setting-out processing of the version information as a stream attribute is performed next. That is, when it judges whether setting out of version information is made in edit preservation processing (step S82) and setting out is made, all the streams in the document structure stored in memory 1 are collected (step S83), and sequential selection of the unsettled stream which omits the following processings is made (step S84).

[0100] And the stream set with which the selected stream belongs is looked for out of document structure (step S85), and version information is added to the attribute of the stream chosen under stream set concerned (step S86). And document preservation processing is ended in the place which ended the processing after the above-mentioned step S84 about all streams (step S87). Therefore, the attribute which it faced saving the stream generated by edit processing, and the stream concerned was added to the same stream set as the stream before edit, and was set up on the occasion of preservation processing is matched with a stream.

[0101] In addition, although the algorithm adopted in the above-mentioned example is choosing serially the stream which suits a selection condition from the stream set currently referred to and generates a document when it meets with support sequentially from root support. After choosing all the streams that suit a selection condition from all the stream sets included in document structure, the these-chosen stream is assembled and you may make it generate a document besides this. Moreover, although the extraction condition is surely specified at the time of starting of a document editor, when there are no directions especially, it is made the same as the last assignment, or may be made to make it default conditions in the above-mentioned example. In this case, if it thinks most that edit of it can be consistently performed by two or more persons in charge as important, default conditions should just make user assignment "% principal." Moreover, what is necessary is just to specify no default conditions, if it thinks that the newest document can be edited as important most.

[0102] Moreover, although a high version was added into document structure on the occasion of edit processing in the above-mentioned example, you may also make it having performed reference processing add a high version. If it does in this way, it will become possible to take out the document structure in the event of performing reference processing.

[0103] Moreover, although premised on criteria time being newer than the time which specified the version in the selection procedure of the stream in above-mentioned this example, when criteria time and version time are reversed, selection processing of the stream can be carried out convenient by taking the following approaches.

1) Give priority to assignment of a version over criteria time.

2) Prevent from specifying criteria time before the time which specified the version. In addition, with a stream attribute, since the time which specified the version is not recorded, for that, a means to record the time which specified the version is required for it.

[0104] Moreover, although version information was added to the stream attribute in the above-mentioned example when a version was specified, you may make it instead add as a new stream. Moreover, although the user is performing assignment of a version clearly, a version may be automatically recorded at the time of a print etc., for example. Moreover, in the above-mentioned example, support names, such as a "claim", are matched with the stream containing support, and are managed. Therefore, although a new stream is generated also by changing the identifier of support, it does not matter even if it manages a support name independently of a stream or makes it manage as an identifier of the stream currently referred to.

[0105] Moreover, although the model which expresses the structure of a document in the unit of a stream as a logical unit group was adopted in the above-mentioned example, this invention is not limited to the expression approach of this document structure, and can also apply this invention to the document structure by the tree structure which consists of a node and a link as shown with the conventional technique. In addition, in the document structure by the node and the link, the link set on the basis of one node corresponds to the logical unit group of this invention. In short, this invention has the description in having given a means to perform the unit of this link set and stream for a duplicate, preservation, attribute setting out, etc. appropriately for every edit processing as an object of version control.

[0106] For example, what is necessary is just to perform it as follows, in order to extend invention indicated by JP,6-35914,A which is the conventional technique and to offer the function of this invention. The link train to a low order node will be treated as a content of the node about the node which is not an end, and it has a link train of the version node corresponding to the node different, respectively. Thus, if it extends, a reference unit [ in / for a low order node / this invention ] and the link train to a low order node can be treated as a logical unit group.

[0107] Furthermore, if it says, in this invention, it will also be possible to adopt the DS shown in the multiplex document processing system system (JP,7-44563,A, JP,6-131343,A) explained as a conventional example as document structure, and the thing corresponding to the logical unit group of this invention will serve as a set of the child node pointer which a node holds in the DS concerned. In addition, although a means to multiplex a node is offered in the multiplex document processing system system concerned, this invention has a means to multiplex a child node pointer set at the time of edit. That is, in this invention, when it edits, it not only multiplexes a node, but it has a means to perform duplicate, preservation, and attribute setting out for the set of the child node pointer. When renewal of a document arises, while specifically

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

reproducing and managing a child node pointer set automatically, it enables it to specify time, a user, and a version name as the attribute value. To say nothing of [ if a user performs such processing manually ] an editing task becoming very complicated, the essence of this invention is by performing this complicated processing automatically and appropriately by the computer to mitigate substantially the burden in the writing activity by two or more persons in charge.

0108] Moreover, although the above-mentioned example showed the example which displays document structure together with the text display of a content block, you may make it display only structure, without displaying the content of a text, as shown in drawing 43 and 44. That is, it is shown by drawing 43 by only the number of candidates displaying the arrow head which shows a stream whether about a rectangle and a stream, support is displayed as an arrow head and there is any candidate of other streams in it. Moreover, it is shown to the right shoulder of the graphic form in which support is shown by drawing 44 by giving a shadow only for the number of candidates whether it is displaying as the tree structure of the support which refers to a stream, and other candidates are in a stream.

0109] Next, one example of the structured data processor concerning this invention is explained. In addition, as compared with the above-mentioned document processing system equipment, a content unit corresponds to the set of the link concerning edit processing that the link and logical unit group of a node and a reference unit are the same (that is, it has a brotherly relation). Moreover, the following explanation about this example also shows the example which generates the structure which did not exist in the past.

0110] Here, this example explains first the variation of structured data made into the object of processing. an intrinsically different point from the DS of the conventional technique which a technique have show the variation of the data-hold approach about the relation between a node, a link, and links, the content of a document, attribute mapping, and a selection attribute to drawing 45 </A> - drawing 47 , and consist of a node and a link be by hold two or more relation between links which make the same node the starting point, and map attribute information in each to manage the relation between links. Also in which variation, it is the constitutionally important requirements for this invention that information required in order to express such structures is managed.

0111] Drawing 45 shows the variation of the implementation approach of the relation between links. The structure shown in this drawing (a) is the configuration of holding two or more relation between links which makes it the starting point during description of a node, and with this structure, since the relation between links does not need to specify a common starting point node, it is realizable as a list of reference pointers to a terminal point node. In addition, in this example, in order to express the current structure held as an object for edit at the current structure management tool 21 so that it may mention later, this configuration is adopted. Moreover, the structure which the relation between links is expressed as a list of reference pointers to a starting point node and a terminal point node, and shows the structure shown in this drawing (b) in this drawing (c) holds the set of the relation between links with a common starting point node as a links relation set, and refer to the starting point node for it from there. In addition, in this example, in order to express the hyper-structure held at the hyper-structure management tool 20 so that it may mention later, the configuration shown in this drawing (c) is adopted.

0112] Drawing 46 shows the variation of the implementation approach of attribute mapping. With the structure shown in this drawing (a), the selection attribute which description of the relation between links mentions later is included. Moreover, with the structure shown in this drawing (b), the links relation set maps the selection attribute in each relation between links, and the management tool of attribute value maps the selection attribute in the relation between links with the structure shown in this drawing (c). In addition, in this example, the structure shown in this drawing (a) is adopted. Drawing 47 shows the variation of the maintenance approach of the content of a document by the node. The structure shown in this drawing (a) is a configuration which describes the content in a node, the structure shown in this drawing (b) is the configuration that description of the relation between links describes the content, and the structure shown in this drawing (c) is the configuration of referring to the content managed with the management tool of the content of a document. In addition, the configuration of this drawing (b) is adopted in this example.

0113] Subsequently, the configuration of the structured data processor concerning this example is shown in drawing 48 , and the structured data of the above-mentioned configuration is processed with this equipment. The structured data processor is equipped with the hyper-structure management tool 20, the current structure management tool 21, the edit means 22, the registration directions means 23, the registration means 24, the hysteresis management tool 25, the hysteresis display means 26, the extract means 27, the extract directions means 28, and the routing means 29.

0114] The hyper-structure management tool 20 has held and managed in memory the structured data which has constituted the graph structure, and this structured data is constituted by the relation between a node, a link, and links, a selection attribute, and attribute mapping. A node is a smallest unit which constitutes structured data, and the node holds the text information which is the content of a document in this example which structured document data. In addition, the tree structure which combined the node by the link is the current structure of expressing a document. A link is a unit showing the point relation between two nodes, and a link is defined as a group of the node used as the node used as the starting point,

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

and a terminal point. In addition, in this example, a link is included in description of "LinkList" which describes the relation between links. Moreover, this hyper-structure management tool 20 constitutes the node maintenance means, link maintenance means, and link relation maintenance means which are told to a claim.

[0115] The relation between links has described the relation to the set of the link which makes the same node the starting point, and specifies the list of links included in this set, and the location of each link. In addition, sequence is prescribed to the link set by this example. A selection attribute is the information corresponding to the selection information of the above-mentioned example, and is the attribute information for choosing the relation between links from one or more relation between links which makes the same node the starting point. In addition, in this example, the list of the date and time of creation, an implementor name, and version names is used as a selection attribute. Attribute mapping is description which relates a selection attribute with the relation between links, and "LinkList" which describes the relation between links holds attribute information in this example.

[0116] The current structure management tool 21 has held and managed in memory the document structure data expressed by the tree structure (a kind of the graph structure) which consists of the relation between a node, a link, and links, and this document structure is set as the object of edit processing. In addition, in this example, description of the relation between links and a link is included during description of "Node" which describes a node. The edit means 22 performs edit processings, such as generation about the relation between the node contained in document structure according to directions of a user, a link, and links, updating, and deletion, and has the candidate display means 30, the candidate directions means 31, and the exchange means 32. The candidate display means 30 is a means to display the relation between links which makes the same node the starting point on a display screen, and displays the relation between links which makes the starting point the node which the user chose in this example. The candidate directions means 31 is a means to choose any one from the relation between links displayed by the candidate display means 30 according to directions of a user. The exchange means 32 is a means to change the relation between links chosen into the relation between links in which it was directed by the user.

[0117] The registration directions means 23 is a means to receive the directions for making the document structure currently held at the current structure management tool 21 register into the hyper-structure management tool 20 from a user, and the registration means 24 registers document structure into hyper-structure according to the directions concerned. In addition, registration is storing the node in current structure, and the relation between links in hyper-structure, and performing suitable attribute information mapping. The hysteresis management tool 25 is a means to manage the registration hysteresis of current structure, and whenever it makes the document structure currently held at the current structure management tool 21 register into the hyper-structure management tool 20, it records and manages registration hysteresis. In addition, in this example, the group of registration time, a registrant name, and the version name set up at the time of registration is recorded as a session. The hysteresis display means 26 is a means to display as a list the information on the session managed with the hysteresis management tool 25 on a display screen.

[0118] The extract means 27 extracts document structure from the hyper-structure management tool 20 using the relation between links chosen according to the directions from a user, is a means to make the document structure concerned hold to the current structure management tool 21, and has the retrieval means 33 and the selection means 34. A retrieval means 33 is a means search the relation between links which fulfills the retrieval conditions directed by the user out of the relation between links which makes the same node the starting point, and a selection means 34 is a means choose the relation between [ of one ] links according to the selection rule directed by the user from the searched relation between links, in it.

[0119] The extract directions means 28 is a means to receive directions of an extraction condition from a user, and has the retrieval condition directions means 35 and the selection-rule directions means 36. In addition, in this example, two, a selection condition and selection rule, can be directed as an extraction condition. The retrieval condition directions means 35 is a means to receive directions of a selection condition (namely, retrieval conditions) from a user, and the selection condition includes time, the version name, and the writer name in this example. The selection-rule directions means 36 is a means to receive directions of selection rule from a user, and selection rule are user precedence, time precedence, or version name precedence in this example.

[0120] The routing means 29 is a means to choose the node used as the root of current structure from the hyper-structures currently held at the hyper-structure management tool 20, in this example, is making sequential selection of the relation between links with a root node as the starting point, and is generating the current structure held at the current structure management tool 21. In addition, each above-mentioned functional means 20-36 consist of this examples, when a processor performs a predetermined control program.

[0121] Here, the above-mentioned extract directions means 28, the hysteresis display means 26, and the edit means 22 have the function which carries out the display output of the interface screen to show information to a user or for a user input directions, information, etc. The extract directions means 28 carries out the display output of the display screen as shown in

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

drawing 49, and each following function part is set to this screen. The root node name input section 40 is the area for inputting the node name used as the root of the document data extracted from the hyper-structure management tool 20, and the retrieval condition input section 41 is the area for inputting a selection condition. The version name section 45 which is the area for inputting the version name of the time sections 43 and 44 which are the area for inputting time with the condition of the user name section 42 which is the area for inputting the identifier of the user who is doing the current activity in order to input the above-mentioned selection condition into this retrieval condition input section 41, and document data to extract, and a document to extract is formed.

0122] The selection-rule input section 46 is area where a user specifies either of the selection rules. Processing which chooses a match preferentially is given to the user specified from that with which the selection means 34 will fill a selection condition if user precedence (USER) is chosen. Processing whose generation time chooses the newest thing from that with which the selection means 34 will fill a selection condition if time precedence (DATE) is chosen preferentially on the basis of assignment time is performed. Selection of version name precedence (VERSION) performs processing which chooses preferentially a thing including the version name specified from that with which the selection means 34 fills a selection condition. Moreover, the writer selection section 47 is the area where a user can choose writers other than a user as a part of selection condition.

0123] Moreover, the "extract" carbon button and the "history" carbon button are prepared in the manual operation button section 48. If a "extract" carbon button is chosen, the processing which extracts document data from the hyper-structure management tool 20 based on a selection condition and selection rule by making the directed root node into the starting point will be started, and the hysteresis management tool 25 will be started by this and coincidence. Moreover, if a "history" carbon button is chosen, the hysteresis display means 26 will be started and the registration hysteresis of the document performed by then will be displayed on a screen. The example of the registration hysteresis in which the hysteresis display means 26 carries out a screen display is shown in drawing 50, and time, a user name, and a version name are displayed on each hysteresis (namely, each line) by which it was indicated by the list as information about one registration processing.

0124] The edit means 22 carries out the display output of the display screen as shown in drawing 51, and a display output is generated and carried out to this screen by selection of the above-mentioned "extract" carbon button. The document structure edit window 50, the content edit window 51 of a document, the candidate selection window 52, the edit carbon button 53, and the extraction condition display 54 are formed in this screen. The document structure edit window 50 can display the structure of the document data extracted by the extract means 27, and a user can choose the node contained in document structure by the pointing device.

0125] A user can update the content of the selected node by editing the content of a text currently displayed by the content edit window 51 of a document displaying the content of the node chosen in the document structure edit window 50. The candidate selection window 52 displays the candidate of the linked list which makes the selected node the starting point, and reverses and shows the linked list by which current selection is made. A user can change for what chooses other linked lists and is chosen with the pointing device now in this window 52.

0126] Each following carbon button is prepared in the edit carbon button 53, and a user can make edit and preservation of a document perform using these carbon buttons. The child node of the node by which current selection is made if a "add child" carbon button is chosen is generated. The twin node of the node which is making current selection if a "add brother" carbon button is chosen is generated. The node which is making current selection if a "move down" carbon button is chosen is moved more on the same level. The node which is making current selection if a "move up" carbon button is chosen is moved more in front on the same level. The level of the node which is making current selection if a "move left" carbon button is chosen is raised one step. The level of the node which is making current selection if a "moveright" carbon button is chosen is lowered one step. The node which is making current selection if a "move node" carbon button is chosen is deleted. If the document which the current structure management tool 21 will hold if a "save" carbon button is chosen is saved in the hyper-structure which the hyper-structure management tool 20 holds and chooses a "close" carbon button, the whole window will be closed and edit processing will be completed. Moreover, the selection condition and selection rule which were specified as conditions for extracting current structure are displayed on the extraction condition display 54.

0127] Next, it explains more concretely using the example which shows processing of the structured data concerning this example to drawing 52. First, the hysteresis of edit activities of the document by the writer of trinomial, Peter, David, and Vicky is shown, the axis of abscissa in drawing shows the passage of time, and the axis of ordinate shows the person in charge to drawing 52. Moreover, (a) - in this drawing (e) shows the structure of the document data saved in the time corresponding to the location of an axis of abscissa by the writer corresponding to the location of the axis of ordinate to the hyper-structure management tool 20 (registration). In addition, drawing 53 shows the hysteresis of edit with the derivation tree which is the conventional technique, the arrow head in drawing expresses derived relation, and derived relation means that the document structure where it was located in a terminal point was made from the document structure where it is

**BEST AVAILABLE COPY**

USPTO  
PAGE BLANK (USPTO)



located at the starting point. Such derived relation constitutes the tree structure which makes a node the condition of each which is called a derivation tree ] document in an event.

0128] By drawing 52, the indented text shows the tree structure of document structure, and the DS expressing the tree structure which (a) - (c) expresses with drawing 54 is shown. Drawing 55 - drawing 57 show the condition of hyper-structure when preservation of the document structure of (a) - (c) is performed. They are this example and the example which is writing the manual of a product called "WriteAtWill" by three persons, Peter, David, and Wicky. The circumstances of later in the activity by order are explained below.

0129] first, it sets to (Aug 1) -- Peter began to write the manual of "WriteAtWill" and saved the document data (a) which are in this first condition by the identifier of "PC version". Subsequently, since two description with DOS and WIN for OS was needed for (Aug 2), the document data (b) which divided \*\*\*\*\* of document data (a) into DOS and WIN were created, and it saved, without naming. Then, it was decided to write two different manuals by the respectively different person in charge, and charge was assigned to David and Wicky, respectively. And the document data (c) with which David took out the description for DOS to (Aug 3) out of document data (a), and corrected it were created, and it saved by the identifier of "DOS version". Furthermore, the document data (e) with which Wicky took out the description for WIN to (Aug 4) out of document data (b), and corrected it were created, and it saved by the identifier of "WIN version". And David created the document data (d) which corrected document data (c) further to (Aug 5), and saved five, without naming.

0130] In such a case, it can set and the document data which did not exist in the past can be generated in this example. Namely, this invention not only can restore the version of the document which existed in the past like the conventional technique, but can take out the structure of the document which did not exist. The document data (g) of drawing 58 and (f) are document data which are not on the derivation tree shown in drawing 53. The structure of this document data (g) is structure which transposes the part below "INTRO" of the structures of (Aug 4) of Wicky to that to which David corrected (Aug 3), and is acquired. That is, document data (g) leave the part which he wrote for Wicky as it is, and make the newest information that other writers wrote reflect. Moreover, document data (f) incorporate the newest information that other writers wrote into the framework of the whole which Peter defined as (Aug 2). Therefore, it is possible to take out simply the new document structure which did not exist in such the past and had not carried out a schedule in this invention, either using the part of the document made in the past.

0131] Here, with reference to drawing 59 which shows the selection condition when editing a document, and the variation of an extract result - drawing 74, processing of a document extract is more concretely explained at the event of (Aug 6). The hysteresis of the document preservation performed before displaying on drawing 59 with the hysteresis display means 26 (Aug 6) is shown, and the selection condition specified by David is shown in drawing 60. Input assignment of "WAW Manual" which is the title of a manual as an identifier of a root node is carried out on the screen shown in drawing 60 at the root node name input section 40. Input assignment of "David" is carried out as a user's identifier at the user name input section 42. Input assignment of "Aug/06 / 96 "time amount [ and ]" 19:03" is carried out. as the information on time which shows whether it searches for the content created by when in the time input section 43 -- the date -- Input assignment of "DOS version" is carried out as a version name which shows the version used as the origin which starts retrieval at the version name input section 45.

0132] In addition, since nobody specifies, the content of a document which David which is a user created is searched with this example by the writer selection section 47 which specifies whether it searches for the content which which writers other than a user created. Moreover, in this example, it is directed that "USER" is chosen in the selection-rule input section 46, and the content created by itself is given priority to and chosen.

0133] The result of having extracted document data from the hyper-structure Management Department 20 according to the conditions shown in drawing 60 is shown in drawing 61. In addition, in this example, the document data (d) shown in drawing 52 are extracted. Moreover, the conditions shown in drawing 60 are changed, and as shown in drawing 62, when the conditions of the time input section 43 are specified as "Aug / 03/96", the extract of the document data reflecting the content newest even by (Aug 3) is directed. Consequently, document structure as shown in drawing 63 is extracted from the hyper-structure Management Department 20, and a screen display is carried out. In addition, this extracted document data is document data (c) shown in drawing 52 in this example.

0134] Moreover, if the conditions shown in drawing 60 are changed, assignment of the user input section 42 is made into "Nicky" as shown in drawing 64, and assignment of the version name input section 45 is made into version name "WIN version", document structure as shown in drawing 65 will be extracted from the hyper-structure Management Department 20, and a screen display will be carried out. In addition, this extracted document data is document data (e) shown in drawing 52 in this example. Moreover, the conditions shown in drawing 64 are changed, as shown in drawing 66, assignment of the selection-rule input section 46 is made into "DATE", if assignment of the writer selection section 47 is further specified that it adds David and Wicky as other writers, document structure as shown in drawing 67 will be extracted from the hyper-

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

structure Management Department 20, and a screen display will be carried out. In addition, this extracted document data is document data (g) shown in drawing 58 in this example, and the DS which was not able to be chosen is extracted with the conventional technique not existing in the derivation tree shown in drawing 52.

0135] Moreover, if assignment of the user name input section 42 is made into "Peter" as shown in drawing 68, and assignment of the version name input 41 is further changed into "PC version", document structure as shown in drawing 69 will be extracted from the hyper-structure Management Department 20, and a screen display will be carried out. In addition, this extracted document data is document data (b) shown in drawing 52 in this example. Moreover, the conditions shown in drawing 68 are changed, if assignment of the selection-rule input section 46 is made into "VERSION" as shown in drawing 70, document structure as shown in drawing 71 will be extracted from the hyper-structure Management Department 20, and a screen display will be carried out. In addition, this extracted document data is document data (a) shown in drawing 52 in this example.

0136] Moreover, the conditions shown in drawing 68 are changed, if assignment of the writer selection section 47 is specified that it adds David and Wicky as other writers as shown in drawing 72, document structure as shown in drawing 73 will be extracted from the hyper-structure Management Department 20, and a screen display will be carried out. In addition, this extracted document data is document data (g) shown in drawing 58 in this example, and the DS which was not able to be chosen is extracted with the conventional technique not existing in the derivation tree shown in drawing 52. Here, on the screen shown in drawing 73, the candidate of the linked list which can be replaced on the further top level is displayed on the candidate selection window 52, and the inverse video of the present selection situation is carried out. In this example, the inverse video shows that the linked list of the version name of "Win version" which Wicky created is chosen from four alternative.

0137] Moreover, what Peter created is chosen as the past from the linked list shown as alternative in drawing 73, and the condition of having performed the directions replaced with the present thing is shown in drawing 74. Consequently, the structure of the document data (f) in drawing 58 is extracted, and the DS which was not able to be chosen is extracted with the conventional technique not existing in the derivation tree shown in drawing 52.

0138] Next, the DS and the algorithm of this example are explained. This example is realized by "Java" (trademark) language on "Windows 95" (trademark). "Java" is an object-oriented language and describes the procedure called the DS called an object and the method which the object received and performs as a definition of a class. The program listing shown below extracts and simplifies main parts from the Java program which has realized this example, and DS and an algorithm are explained, referring to this program listing. In addition, DS is shown in drawing 75.

0139]

```
1) Class definition of set "LinkedListHolder" of links relation description "1" public class LinkedListHolder extends Object
//LinkedListHolder is an object which manages the set of the linked list which has the same node at the starting point. /*
Definition of DS */2" Node sourceNode; The node used as//starting point "3" Vector linkListSet; A set/* of//linked list
Definition of Method registerLinks */4" void registerLinks (Node node, String user, Date date, String version) {-- The method
which registers the linked list which makes the starting point // specified node into a linked list set (addition) [0140]
5" int index =-1; The variable which stores the number of the same linked list under// linked list set "6" LinkedList linkList; The
variable which stores the same linked list under// linked list set "7" for (int i=0; i<linkListSet.size();i++) {-- "8" which looks for
from ] the same linked list while registering with// linked list set linkList =; (linkListSet(linkList).elementAt (i)) "9" if
linkList.isLinksFor (node)) {"10" index= i; "11" break; "12" } "13" } "14" if (index >= 0) It is [0141] when (// the same linked list
. found.
15" if (version.length() == 0) {-- Whether// version name was specified A check "16" } else {"17" linkList.addVersion
(version); A version name in// found link set An addition "18" } "19" }else {-- When// the same linked list is not found, "20"
linkList = new LinkedList (node, user, date, version); It is generation [0142] about the linked list which matched a user name,
me, and a version name with the linked list which //node has.
21" addLinkedList (linkList); // newly generated linked list is added. "22" } "23" }/* Definition of Method selectLinks */24" Vector
selectLinks (String user, Vector authors, Date date, String version, int rule) {-- Method which chooses one according to the
regulation (rule) specified from the list of links which fulfills//extraction condition. Since LinkedListSet is registered so that it may
link with the new order of registration time, that which was found first becomes the newest linked list. "25" int userindex =-1;
variable which stores the number of the first linked list with which//user name agrees [0143]
26" int dateindex =-1; The variable which stores the number of the first linked list with which the conditions of//time agree
27" int versionindex =-1; The variable which stores the number of the first linked list with which// version name agrees "28"
for (int i=0; i<linkListSet.size(); i++) {-- All the linked lists under// linked list set Retrieval "29" LinkedList linkList =(LinkedList)
linkListSet.elementAt (i); "30" if (linkList.satisfies (authors, date, version)) It is activation [0144] about the following about
that fulfills the conditions about { // writer, time, and a version name.
```

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

```

"31" if (dateindex < 0) {-- //If what still fulfills the conditions of time is not found, "32" dateindex = i; //"33" which makes this
linked list the newest thing } "34" if (linkList.regUser.equals (user)) {-- //It will be "35" if there is a thing corresponding to a
user name. if (userindex < 0) { //It is [0145] if what still agrees in a user name is not found.
"36" userindex = i; //"37" which makes this linked list the newest thing } "38" } "39" if (linkList.hasVersion (version)) {-- //It will
be "40" if there are some with which a version name agrees. if (versionindex < 0) { //It is [0146] if that with which a version
name agrees is not found.
"41" versionindex = i; //"42" which makes this linked list the newest thing } "43" } "44" } "45" } [0147]
"46" int index = -1; //The variable which stores the number of the linked list to choose "47" if (rule == DATE) {-- //When
selection rule are the time newest, "48" index = dateindex; //Time the newest thing Selection "49" } else if (rule == USER) {--
//the time of selection rule being user name precedence -- "50" if (userindex < 0) [if there is nothing corresponding to // user
name -- 0148]
"51" index = dateindex; //The newest thing Selection "52" } else {-- //If there is a thing corresponding to a user name, "53"
index = userindex; //"54" which chooses the newest thing corresponding to a user name } "55" } else if (rule == VERSION) {--
//[the time of selection rule being version name precedence -- 0149]
"56" -- if there is no selected thing -- "59" return new Vector(); //-- "60" } else {[0150 which returns the list of empty -- ] index
= versionindex; //"57" which chooses the newest thing with which a version name agrees } "58" if (index < 0) {-- // [ ]
"61" return.(linkListSet(LinkList).elementAt (index)) links; //"62" which returns the selected linked list } "63" } "64" } [0151]
2) Class definition of the node description Node "65" public class Node extends Object { //The object / * DS */"66" String
contents which describes a node; //Content of text "67" Vector links which a node holds; //List "68" Long nodeID of the link
which makes a node the starting point; //Identifier of a node. It is considered that the node whose value of this corresponds
to the same node. "69" } [0152]
3) class definition of the links relation description LinkList "70" public class LinkList extends Object {-- //The object/* showing
the list of links which make the same node the starting point Definition of DS */"71" String sourceNodeContents; //The
content of the source node "72" Vector links; //The list of terminal point nodes "73" Date regDate; //Date and time of creation
"74" String regUser; //Implementer "75" Vector versions; //List of version names [0153]
Definition of Method isLinksFor */"76" public boolean isLinksFor (Node node) { //Method which investigates whether it is in
agreement with the linked list of the specified node. "77" if (!sourceNodeContents.equals (node.contents)) return false; //"78"
which is not in agreement if the content differs from a source node if (links.size() != node.links.size()) return false; //"79"
which is not in agreement if the die length of a linked list is different boolean equal = true; //"80" which also investigates
whether the number of conflicting things is one for (int i=0; i < links.size(); i++) {[0154]
"81" equal = (links(Node).elementAt (i)) equals (Node) (node.links.elementAt (i)); "82" if (!equal) break; "83" } "84" return
equal; "85" } [0155]
Definition of Method satisfies */"86" public boolean satisfies (Vector authors, Date date, String version) {"88" return false;
which will not be filled if it is the back [ time / as which method "87" if (regDate.after (date)) { // registration time which
investigates whether // conditions are fulfilled were specified } "89" } [0156]
"90" for (int i=0; i<versions.size(); i++) {-- //"91" which will be filled if the specified version name is included if (.(versions
tring).elementAt (i)) equals (version)) {"92" return true; "93" } "94" } "95" for (int i = 0; i < authors.size(); i++) {-- //It will fill, if
a specified writer is included. [0157]
"96" String au = authors(String).elementAt (i); "97" if (regUser.equals (au)) {"98" return true; "99" } "101" return false; "102" }
"103" } [0158] } "100"
4) class definition of hyper-structure management tool HyperStructureManager "104" public class HyperStructureManager
extends Object {-- //The object/* which manages the whole structure which consists of a node and a link Definition of DS
"105" Vector linkListHolderSet; //The set of the linked list contained "106" Vector regVersions; //The list of version names
registered "107" Vector regAuthors; //List of author names registered "108" Vector sessionHistory; //Hysteresis of the
formed register operation [0159]
Definition of Method saveDocument */"109" void saveDocument (Node root, String user, Date date, String version) {--
the method which registers the current structure represented by the root node "110" if (root == null) {"111" else if
gVersionsHas (version)) { //Nothing will be carried out if the version name is already registered.} "112" } else {"113" if
ersion.length() > 0) { //If the version name is specified, it is "114" regVersions.addElement(version); //. "115" } which
registers the directed version name [0160]
"116" saveTree (root, user, date, version); //"117" which registers structure traversable from a root node into hyper-structure
sessionHistory.addElement (new Session (user, date, version)); //It is an addition "118" about the hysteresis of register
eration. } "119" } [0161]
Definition of Method saveTree */"120" void saveTree (Node node, String user, Date date, String version) {-- //The method

```

BEST AVAILABLE COPY

BLANK (SPTD)

which registers structure traversable from the directed node into hyper-structure "121" int index = -1; //"122" which looks for the linked list electrode holder which makes a node the starting point from hyper-structure for () [ int ] i = 0; i < linkListHolderSet.size(); i++ {"123" if (linkListHolderSet(LinkListHolder).elementAt (i)) sourceNode.equals (node)) {"124" index = i; "125" break; [0162] "126" } "127" } "128" LinkListHolder llholder; //The variable which stores the linked list electrode holder which makes the starting point the directed node "129" if (index < 0) {-- //It is "130" if a linked list electrode holder is not found. llholder = new LinkListHolder (node); //[0163] which generates and chooses a new linked list electrode holder "131" linkListHolderSet.addElement (llholder); //"132" which adds the generated linked list electrode holder } else {"133" llholder = linkListHolderSet(LinkListHolder).elementAt (index); //If found, the linked list electrode holder will be chosen. "134" } "135" llholder.registerLinks (node, user, date, version); //[0164] which adds the linked list of the node directed in the linked list electrode holder "136" for (int i = 0; i < node.links.size(); i++) {"137" saveTree (Node) (node.links.elementAt (i), user, date, version); //"138" } which calls a saveTree method recursively about the node of the terminal point of the link of the directed node "139" } [0165]

\* Definition of Method extractDocument \* //"140" void extractDocument (Node root, String user, Vector authors, Date date, String version, int rule) {-- //The method which extracts the current structure of expressing a document with the node as the starting point specified as a root node "141" LinkListHolder linkListHolder = linkListHolderFor (root); //"142" which chooses the linked list electrode holder which makes a root node the starting point root.contents = linkListHolder.sourceNode.contents; //"143" which reads the content of the linked list electrode holder as a content of the root node Date sesdate = sessionDateOf (version); //"144" which takes out the time into which the version name was registered if (sesdate != null) {"145" if (date.before (sesdate)) {/[0166] to which the specified time investigates \*\*\*\*\* before "146" } else {"147" extractTree (root, user, authors, date, version, rule); //"148" which extracts structure with a root node as the starting point at the time of the appointed day after a registration day } "149" } "150" } [0167]

Definition of Method extractTree \* //"151" void extractTree (Node node, String user, Vector authors, Date date, String version, int rule) {-- //The method which extracts structure with the specified node as the starting point "152" LinkListHolder llholder = linkListHolderFor (node); //"153" which chooses the linked list electrode holder which makes the starting point the specified node Vector links = llholder.selectLinks (user, authors, date, version, rule); //"154" which chooses the link which fills an extraction condition from linked list electrode holders node.links = new Vector(); //It is initialization "155" about the linked list of a node. Node nd = null; [0168] "156" for (int i=0; i < links.size(); i++) {-- //The following about each link of the selected linked list Activation "157" nd = (links.elementAt (i)) copy(); //"158" which copies the terminal point node of a link and generates a new node node.addNode (nd); //"159" added to the linked list of the node which had the generated node specified extractTree(nd, user, authors, date, version, rule); //It starts from the generated node. extractTree is performed recursively. "160" } "161" } [0169] Moreover, the above of the main components of this example is shown in drawing 75 . The current structure shown in this drawing (a) is the structure expressing the document for edit, and is expressing this current structure at this example in the form of the tree structure which consists of node description "Node". The DS of this "Node" is the definition of the object to which from the 65th line of the above-mentioned list to the 68th line corresponds, and includes the next description. "contents" has the content of a document and is a character string in this example. "links" has the list of reference to one or more "Node", and this list is a relation description which defines a set and order relation of the link which makes this "Node" the starting point. "id" has the identifier of a node and it is judged with this value whether a node is the same.

70] The hyper-structure shown in drawing 75 (b) is structured data by which maintenance management is carried out with the hyper-structure management tool 20, and is expressing this hyper-structure in the form of the network structure of the hyper-structure structured by "LinkList" and "LinkListHolder" at this example. Although the structure of document data is constituted by the reference to other "Node" by "links" of "Node", it is constituted out of hyper-structure by the reference from "LinkList" and "LinkListHolder". This hyper-structure includes the next description. "LinkList" is links relation description which describes the set of a link and the relation in the meantime which make the same node the starting point, and also describes further the selection attribute matched with this links relation description. In addition, it is the definition of the object which from the 70th line of the above-mentioned list to the 75th line corresponds.

71] Moreover, "sourceNodeContents" has the content of a document of a node used as the starting point of a link, and "links" has the list of reference to Node used as the terminal point of a link. Moreover, "regDate" has the date and time of registration which is a selection attribute, "regUser" has the implementor name which is a selection attribute, and "versions" has the list of version names which are selection attributes. "LinkListHolder" is the definition of the object to which from the 1st line of the list which held the set of LinkList which makes the same Node the starting point, and was described above to the

BEST AVAILABLE COPY

PAGE BLANK (USPTO)



```

//aipn1.ipdl.jpo.go.jp/cgi-bin/tran_web.cgi ejje

```

US  
SAGE BLANK (USPTO)

diagram -- containing -- as -- changing -- having saved . furthermore -- (-- Aug -- -- three --) -- Peter -- (-- c --) -- being shown -- as -- (-- a --) -- being new -- a graphic element -- it is -- a character string -- " -- count -- the section -- " -- adding -- having saved .

[0179] (a) of the above [ drawing 77 ], (b), and (c) -- each DS is shown and a different point from the above-mentioned example is that had a graphic element as a content of a document, and the relation between links has specified the 2-dimensional location as a reference location in a graphic element. By each drawing (a), (b), and (c), (1) illustrates the logical structure of data and it illustrates (2) including relation with the graphic element currently used in order to describe the relation during a link, while it is the content of a document. moreover -- (a) of drawing 78 -- hyper--- a paper mulberry -- the DS in the hyper-structure held at a management tool 20 is shown. Moreover, the selection condition is shown in (b) of this Drawing, and by directing the selection condition concerned, the graphic form shown in (a) of drawing 79 , (b), and (c) is extracted as current structure, and is held at the current structure management tool 21. In addition, the extracted DS which s shown here is the structure which incorporated the newest condition of edit of both Peter and David, and did not exist in in he past.

[0180] Next, one example of further others which applied the structured data processor of this invention to retrieval of a hypertext is explained. In addition, although the points made into the object of the processing of the tree structure generated is a search history instead of edit differ in this example, essential DS and an essential algorithm are the same as the above-mentioned example.

[0181] An example of the network retrieval which Peter performed to (Aug 1) for a certain examination is shown in drawing 80 . That is, first, the page of the address called <http://AAA.html> was opened, and the support to the page BBB related from there was chosen and jumped. Consequently, the page of the address called <http://BBB.html> chose and jumped an aperture and the support to the page DDD further related from there. Consequently, the page of the address called <http://DDD.html> carried out an aperture and the directions which return from there to a front page further. Consequently, the page of the address called <http://BBB.html> chose an aperture and the support to the page EEE further related from there, it jumped, and the page of the address called <http://EEE.html> opened as that result.

[0182] An example of the network retrieval which David performed to (Aug 2) for examination of the same object is shown in drawing 81 . That is, first, the page of the address called <http://AAA.html> was opened, and the support to the page BBB related from there was chosen and jumped. Consequently, the page of the address called <http://DDD.html> carried out an aperture and the directions which return from there to a front page further. Consequently, the page of the address called <http://BBB.html> chose and jumped an aperture and the support to the page CCC further related from there. Consequently, the page of the address called <http://CCC.html> is an aperture, The support from there to the related page FFF was chosen and jumped further. Consequently, the page of the address called <http://FFF.html> carried out an aperture and the directions which return from there to a front page further. Consequently, the page of the address called <http://CCC.html> chose an aperture and the support to the page GGG further related from there, it jumped, and the page of the address called <http://GGG.html> opened as this result.

[0183] In (a) of drawing 82 , and (b), the hysteresis of each above-mentioned retrieval is expressed as the tree structure. The tree structure showing the searched data space is generated by making into the root the node which started retrieval and choosing only the link used for retrieval. In addition, the conventional technique indicated by JP,6-259312,A is adopted as an algorithm which acquires such the tree structure from the search history of a hypertext. In addition, the result registered into hyper-structure is shown in (c) of this drawing by making structure of the hysteresis of retrieval into current structure. In addition, in this example, hyper-structure is acquired combining two or more substructures of the network structure distributed [ WWW ]. As information for the client which performs a search, hyper-structure may be made to be held locally, and it distributes to the countless server which is maintaining the network, and it may be made to be held. [0184] The selection condition specified as (a) of drawing 83 in order to extract current structure from the hyper-structure management tool 20 is shown, and the extracted current structure is shown in (b) of this drawing. This extracted current structure is the structure which did not exist in the past showing the retrieval space of the whole examination which merged examination which Peter and David conducted.

[0185] In addition, this invention can manage both the version of components, and the version of whole structure in addition to the above-mentioned example, and can also apply it to various applications combining flexibly has a meaning. Moreover, the structure which could be made to reflect without constraint of the new information written with each view, and was enabled with a new situation taking advantage of the fragment of the past information structure is generable by applying this attention. For example, the structure of new information can be built from restoration of the past argument, and the fragment of the past argument by applying to opinion exchange with the dispersed electronic mail and dispersed electronic bulletin board in an environment.

[0186] Moreover, even if it corrects direct purport of a letter by the comment or correction by applying to the system which

**BEST AVAILABLE COPY**

**THIS PAGE BLANK (USPTO)**

manages many documents so that the person who wrote does not know each other, trouble does not happen, but a document database with the function which compounds a still newer document can be realized. Moreover, the function to perform restoration of the flow of a request of the past work, analysis of the operation using it, and a design is realizable by applying this invention to management of the structure of flow, such as an electronic mail, the structure of an organization, etc. Moreover, not only when there are two or more persons in charge, but even when it applies to the system into which a document is edited by one person, the function which extracts a suitable part from the document made by then, and is reconfigured in a required document for the object which was not assumed at the beginning can be realized.

0187]

Effect of the Invention] As explained above, according to the structured data processor concerning this invention About the structured data of the graph structure which consisted of a node and a link Since it was made to make the physical relationship information during the link included during each set hold while making the link concerning the same edit processing of two or more links which make the starting point the node same for a links relation maintenance means hold as a set, Even if it makes a user edit freely about the structured data which makes the graph structure of document data etc., creation of the suitable DS doubled with restoration and the new situation of the past DS is realizable. Moreover, it is realizable that this invention makes the suitable structured data set by the new situation which carried out based on the data made in the past, and was not assumed by simple assignment actuation by the user at the beginning.

0188] Even if it is when the whole document structure is changed by edit since a layered structure constitutes document structure in a logical unit group including a logical unit, a new logical unit group is generated on the occasion of edit processing and it was made to hold as the same set as the original logical unit group according to the document processing system equipment and the approach of starting this invention especially, a document can be restored with the structure before edit. Therefore, even if it makes a writer edit a document freely, without giving the constraint on edit, the past document can be restored without trouble, and co-authoring can be realized without trouble. Moreover, according to the document processing system equipment and the approach concerning this invention, since selection information was matched with each logical unit group, by simple actuation of specifying the selection condition containing selection information, a user can restore the document version of the suiting past and can perform reference and edit processing easily about the document of the version of arbitration.

---

translation done.]

**BEST AVAILABLE COPY**

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-171510

(43) 公開日 平成9年(1997)6月30日

(51) Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 17/27

17/21

G 0 6 F 15/20

5 5 0 E

5 7 0 R

5 7 0 D

5 9 6 B

審査請求 未請求 請求項の数21 F D (全 56 頁)

(21) 出願番号 特願平8-297425

(22) 出願日 平成8年(1996)10月18日

(31) 優先権主張番号 特願平7-296154

(32) 優先日 平7(1995)10月19日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 林 浩一

神奈川県足柄上郡中井町境430 グリーン

テクなかい 富士ゼロックス株式会社内

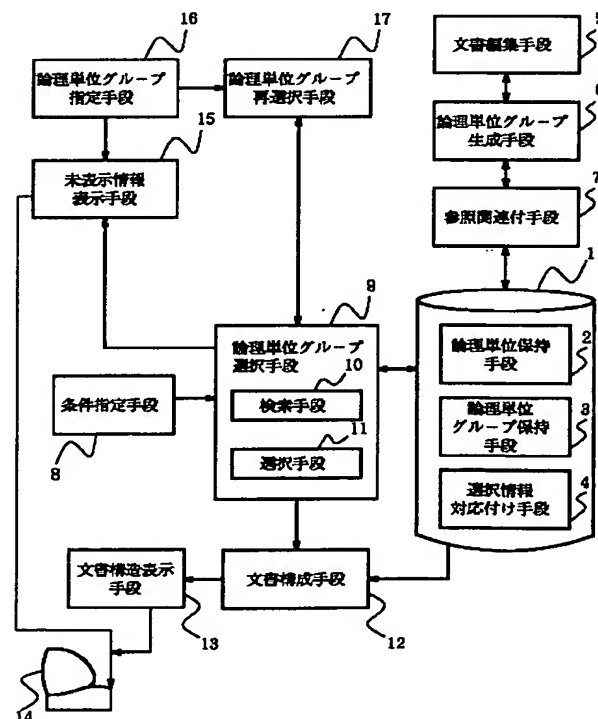
(74) 代理人 弁理士 守山 辰雄

(54) 【発明の名称】 構造化データ処理装置

(57) 【要約】

【課題】 執筆者に編集上の制約を与えずに文書を自由に編集させても、過去の文書の復元や過去には存在していない文書の作成等を可能にして、共同執筆を支障無く実現させる。

【解決手段】 構造化データ処理を構造化文書データに適用した文書処理装置では、論理単位保持手段2に木構造を成している構造化文書の内容の一部である内容部又は下位階層の構造との関連位置を表すアンカーの少なくともいずれか一方を含む論理単位を保持させ、論理単位グループ保持手段3に構造化文書を構成する木構造の同じ階層に位置する論理単位から成るストリームを保持させる。そして、文書編集手段5によって文書構造を編集すると、論理単位グループ生成手段6が当該文書編集に対応するストリームを新たに生成し、参照関連付手段7が新たに生成されたストリームと所定のアンカーを関連付ける。この結果、新たに生成されたストリームは既存の構造化文書中に組み入れられ、文書編集に応じた構造化文書が作成される。



**【特許請求の範囲】**

**【請求項1】** グラフ構造を成している構造化データのノード情報を保持するノード保持手段と、  
構造化データのノード間の関連情報を表すリンク情報を保持するリンク保持手段と、

同一のノードを始点とする複数のリンクの内の同一の編集処理に係るリンクを集合として保持するとともに各集合中の含まれるリンク間の位置関係情報を保持するリンク間関係保持手段と、

リンク関係保持手段に保持されたリンク集合中のいずれか1つの集合を特定することによりノード保持手段及びリンク保持手段に保持された情報に基づいて編集対象となる構造化データを特定するカレント構造管理手段と、カレント構造管理手段により特定された構造化データを編集する編集手段と、  
編集処理により変更されたリンクを含むリンク集合を生成してリンク間関係保持手段に保持させる登録手段と、を備えたことを特徴とする構造化データ処理装置。

**【請求項2】** 前記編集手段はノードを追加する処理も行い、また、前記登録手段は追加されたノードをノード保持手段に保持させる処理も行うことを特徴とする請求項1に記載の構造化データ処理装置。

**【請求項3】** 編集処理に係る情報として選択属性を編集処理により変更されたリンク集合に付与する選択属性対応付け手段を、更に備えたことを特徴とする請求項1又は請求項2に記載の構造化データ処理装置。

**【請求項4】** 選択属性はバージョン情報であり、選択属性対応付け手段は編集処理に係る構造化データ中の他のリンクに対してもバージョン情報を付与することを特徴とする請求項3に記載の構造化データ処理装置。

**【請求項5】** 構造化データを再構成させるための選択条件を入力する条件指定手段と、  
入力された選択条件に対応する選択属性が付与されているリンク情報をリンク間関係保持手段から選択するリンク集合選択手段と、  
選択されたリンク集合に基づいて構造化データを再構成するデータ構成手段と、  
を更に備えたことを特徴とする請求項3又は請求項4に記載の構造化データ処理装置。

**【請求項6】** リンク集合選択手段は、入力された選択条件に対応するリンク集合が複数存在する場合には、予め設定された選択規則に従っていずれか1つのリンク集合を選択し、  
データ構成手段は当該リンク集合に基づいて構造化データを再構成する、ことを特徴とする請求項5に記載の構造化データ処理装置。

**【請求項7】** 選択規則は特定の利用者情報が付与されているリンク集合を優先的に選択するものである、ことを特徴とする請求項6に記載の構造化データ処理装置。

**【請求項8】** 選択規則は最新の日時情報が付与されてい

るリンク集合を優先的に選択するものである、ことを特徴とする請求項6に記載の構造化データ処理装置。

**【請求項9】** 選択規則は特定のバージョン情報が付与されているリンク集合を優先的に選択するものである、ことを特徴とする請求項6に記載の構造化データ処理装置。

**【請求項10】** リンク間関係保持手段に保持されている各リンク集合を識別するための情報を表示出力する表示出力手段と、

選択されたリンク集合に基づいてデータ構成手段により再構成された構造化データを表示出力するデータ出力手段と、

を更に備えたことを特徴とする請求項5に記載の構造化データ処理装置。

**【請求項11】** 選択属性対応付け手段によりリンク集合に選択属性が付与される毎に選択属性の内容を履歴情報として保持する履歴保持手段と、  
保持された履歴情報の一覧を表示出力する履歴表示手段と、

を更に備えたことを特徴とする請求項3に記載の構造化データ処理装置。

**【請求項12】** 木構造を成している構造化文書の内容の一部分である内容単位又は直下の階層の構造との関連位置を表す参照単位の少なくともいずれか一方を論理単位として保持する論理単位保持手段と、

構造化文書を構成する木構造において兄弟の関係にある全ての論理単位から成る論理単位グループを保持する論理単位グループ保持手段と、を備え、

前記参照単位は、該参照単位の直下に位置する少なくとも1つの論理単位グループの集合を表すものであり、更に、

文書構造を編集する文書編集手段と、

該文書編集手段によって文書が編集されたことに基づいて、対応する論理単位グループを新たに生成する論理単位グループ生成手段と、

前記論理単位グループ生成手段により生成された論理単位グループと所定の参照単位を関連付ける参照関連付手段と、

を備えたことを特徴とする文書処理装置。

**【請求項13】** 前記参照関連付手段は、前記論理単位グループ生成手段によって生成された論理単位グループを、編集前の論理単位グループと同一の集合に追加することを特徴とする請求項12に記載の文書処理装置。

**【請求項14】** 木構造を成している構造化文書の内容の一部分である内容単位及び直下の階層の構造を表す参照単位を保持する論理単位保持手段と、

構造化文書を構成する木構造において兄弟の関係にある全ての内容単位あるいは参照単位からなる論理単位グループを保持する論理単位グループ保持手段と、を備え、

前記参照単位は、該参照単位の直下に位置する少なくとも



も1つの論理単位グループの集合を表すものであり、更に、前記論理単位グループと該論理単位グループを選択するための選択情報とを対応付ける選択情報対応付け手段と、文書構造を構成するために前記選択情報を含む選択条件を指定する条件指定手段と、前記条件指定手段によって指定された選択条件と前記選択情報対応付け手段によって対応付けられている選択情報に基づいて、前記論理単位グループ保持手段に保持されている論理単位グループを選択する論理単位グループ選択手段と、前記論理単位グループ選択手段により選択された論理単位グループに基づいて文書を構成する文書構成手段と、を備えたことを特徴とする文書処理装置。

【請求項15】 前記選択情報対応付け手段は、選択情報として文書のバージョン情報を前記論理単位グループ保持手段に保持された論理単位グループに対応付けられることを特徴とする請求項14に記載の文書処理装置。

【請求項16】 前記選択情報対応付け手段は、選択情報として文書の更新日時情報を前記論理単位グループ保持手段に保持された論理単位グループに対応付けられることを特徴とする請求項14に記載の文書処理装置。

【請求項17】 前記選択情報対応付け手段は、選択情報として文書の利用者情報を前記論理単位グループ保持手段に保持された論理単位グループに対応付けられることを特徴とする請求項14に記載の文書処理装置。

【請求項18】 前記選択情報対応付け手段により前記論理単位グループに対応付けられる選択情報は、バージョン情報、更新日時情報、利用者情報であり、前記条件指定手段により指定される選択条件は文書のバージョンと日時情報と利用者情報を含むものであり、前記論理単位グループ選択手段は、前記条件指定手段により指定されたバージョン情報に対応する更新日時情報以降で、指定された日時情報以前の更新日時情報と対応付けられ、且つ、指定された利用者情報に一致する論理単位グループを検索する検索手段と、該検索手段による検索の結果、複数の論理単位グループが存在した場合にいずれか1つの該論理単位グループを選択する選択手段と、を有し、前記論理単位グループ選択手段により選択された論理単位グループから構成された文書構造を表示する表示手段を有することを特徴とする請求項14に記載の文書処理装置。

【請求項19】 前記選択手段は、前記検索手段による検索の結果、論理単位グループが存在しなかった場合に、指定されたバージョン情報に一致する論理単位グループを選択することを特徴とする請求項18に記載の文書処理装置。

【請求項20】 構造化文書の内容の一部分である内容

単位又は直下の階層の構造を表す参照単位の中から編集対象を指定するステップと、指定した内容単位又は参照単位を編集するステップと、該編集に対応して、構造化文書を構成する木構造において兄弟の関係にある全ての内容単位あるいは参照単位から成る論理単位グループを生成するステップと、を含み、

前記参照単位は、該参照単位の直下に位置する一つ以上の論理単位グループの集合を表すものであり、更に、該論理単位グループを選択するための選択情報を入力するステップと、

入力された選択情報と新たに生成された論理単位グループを関連付けるステップと、

該論理単位グループを前記指定した内容単位あるいは参照単位を含む論理単位グループの集合に追加するステップと、を含むことを特徴とする文書処理方法。

【請求項21】 文書の内容の一部分である内容単位と直下の階層の構造を表す参照単位を有する構造化文書を選択するための選択条件を入力するステップと、

前記構造化文書を構成する木構造において兄弟の関係にある全ての内容単位又は参照単位からなる論理単位グループの中から、入力された選択条件と論理単位グループに対応付けられている選択情報とに基づいて論理単位グループを選択するステップと、を含み、

前記参照単位は、該参照単位の直下に位置する少なくとも一つの論理単位グループの集合を表すものであり、更に、

選択された論理単位グループより文書を構成するステップと、

構成された文書を表示するステップと、を含むことを特徴とする文書処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、文書データ等といったグラフ構造を成す構造化データを作成、編集、保存等する処理装置及び方法に関し、特に、執筆者にデータ編集上の制約を与えずに自由に編集することを許しても、過去の構造化データの復元や共同執筆を支障無く行うことのできる構造化データ処理装置及び方法に関する。

【0002】

【従来の技術】一般に、ファイル等といった文書情報のまとまりに更新が繰り返された場合に、特定の条件に合致する過去の状態の文書を取り出すことを保証する技術はバージョン管理と呼ばれ、過去の文書の状態はバージョンと呼ばれる。このようなバージョン管理は過去の状態の文書を復元できるという効用に加え、複数の担当者による共同執筆作業を支援できる等の効用がある。

【0003】ここで、従来では、バージョン管理はファイル等の情報のまとまりを単位にして行われており、フ

ファイルを共同編集するときには、複数の担当者が同じファイルを同時には編集できないように制限する等の管理が行われていた。しかしながら、非常にサイズの大きな文書を執筆する場合や、複数の担当者が文書を執筆する場合には、章や節等といった部分単位で文書を編集する方法が便利である。そこで、文書を構造化して部分毎のバージョンを管理して、各担当者が分担を決めて執筆するように取り決めたり、同一の部分を複数の担当者が同時に編集できないように制限を加えることによって、複数の担当者による共同執筆を可能にした発明がなされている。

【0004】従来における構造化した文書のバージョン管理を、図84～図89を参照して更に詳しく説明する。文書の構造は順序付きの木構造として表現され、章や節等といった部分単位を表すノードが親子関係を示すリンクを用いて構造化され、この木構造の末端に内容（章や節等といった部分単位の内容データ）を含んだノードが接続される。すなわち、図84に示す文書構造の例では、ノードAを文書ルートとすると、ノードBは章、ノードC及びDが節に該当し、末端のノード（u）、（v）、（x）がテキスト等のデータ内容を含む内容部となる。なお、章や節等といった部分単位は文書全体としての構造によって定義されているので、文書の全体構造が変化してしまうとバージョン管理の単位が安定しなくなってしまう。そこで、このような従来における文書構造のバージョン管理では、文書の全体構造を固定し、内容部についてバージョン管理を行っている。

【0005】そして、新たな内容部を作成して既存の内容部と入れ替えるといった編集処理を行う場合には、図85に示すように、新たな内容部（y）を既存の内容部（x）と入れ替えるとともに、古い内容部（x）は元の位置に対応付けて記憶管理する。これによって、当該文書の過去の状態（すなわち、編集前の状態）を復元するときには、古い内容部（x）を新しい内容部（y）と再び入れ替えるだけで、所期の目的を達成することができる。

【0006】このようなバージョン管理では、上記した理由から文書の全体構造を変更した場合には、これを管理することができず、上記のように古い内容部を新しい内容部と入れ替えるだけでは過去のバージョンを復元することはできない。例えば、図85に示した元の文書に対して、図86に示すように新たな部分単位の内容部（z）を追加して対応する新たな部分構造（ノード）Eを文書構造に挿入した場合、図87に示すように内容部分（x）に係る部分構造（ノード）Dを削除した場合、図88に示すように部分構造（ノード）Cと部分構造（ノード）Dとの順序を入れ替えた場合、図89に示すように内容部分（x）に係る部分構造（ノード）Dの位置を移動させた場合等においては、文書の全体構造が変更されるため、内容部を入れ換えるだけでは過去の状態

の文書を復元をすることはできない。

【0007】なお、特開平6-35914号公報には、図85に示したような内容部のバージョン管理を利用して、同一の文書に対する共同執筆を支援する文書作成システムが記載されている。この文書作成システムでは、複数の執筆担当者によって同じノードの内容部を編集したときに、それぞれの担当者が編集した内容部に対応させてバージョンノードを作成し、それぞれの内容部をバージョンノードとの対応付けで管理する。更に、それぞれのバージョンノードにはアクセス権を設定しておき、執筆担当者が文書構造中の内容部を編集する際には、バージョンノードに基づいて、その担当者に適合するアクセス権が設定されている内容部を提示してその編集を可能にしている。

【0008】また、特開平7-44563号公報や特開平6-131343号公報には、文書の構造をノードとリンクからなる木構造で表現し、文書内の互いに内容が異なる複数の部分について論理構造関係が同一の場合には、ノードを多重化することによりこれら部分の論理構造を複合して1つの複合文書構造として保持し、更に、それぞれの部分構造を区別するための属性を多重化したノードに付与しておくことにより、複数の木構造の中から必要な文書構造に対応する木構造を選択することができる多重文書処理システムが記載されている。この多重文書処理システムによれば、文書構造を全体構造と内容に単純に分割するよりも柔軟に表現することができ、更に、必要に応じて複数の木構造の中から必要な文書構造に対応する木構造を選択することができる。

【0009】ここで、バージョン管理技術は、文書データの編集以外にも、ソフトウェア開発、CADデータベースなど様々な分野で利用されている技術であり、一般のバージョン管理技術においても、それぞれの部品（すなわち、部分データ）がバージョンを持った複雑なデータの全体構造についてのバージョン管理は重要な課題である。

このようなバージョン管理に関する従来技術としては、例えば、下記のような種々な技術が知られている。

【0010】複雑な構造を持つCADデータベースのバージョン管理技術について網羅的に記述した“R. Katz, 'Toward a Unified Framework for Version Modeling in Engineering Databases', ACM Computing Surveys, vol. 22, no. 4, 1990”では、上述した全体構造のバージョンをコンフィギュレーションと呼んで、部品のバージョンとは別に管理する技術が示されている。この文献の中には様々なバージョン管理技術が示されているが、いずれもバージョン管理の目的は過去の状態を復元することである。そのための方法として、導出木を用いる方法とレイヤを用いる方法が示している。導出木を用いる方法は、変更の履歴としてどのバージョンのものからどのバージョンが派生したかという関係で導出木（derivation

tree)を管理し、その中から一つの状態を選択することによって、過去の状態を復元する。このように導出木を利用して、一つ前の状態からの差分だけを記録することによって、バージョン管理の実現に必要な記憶容量を軽減させることができる。なお、"Sten Minor & Boris Magnusson, 'A Model for Semi-(a)Synchronous Collaborative Editing, proceedings of ECSCW '93, pp. 219-231, 1993"では、導出木による全体構造のバージョン管理を文書編集に応用した例を示している。

【0011】レイヤによる方法では、変更を行ったときに、生成された部品や更新された部品をレイヤというまとまりとして保持し、適切にレイヤを重ね合わせることで過去に存在していた状態を表現できるようにする。重ね合わされた複数のレイヤ中に同じ部品のバージョンが含まれているときには、後から重ねたレイヤの部品を選択することによって、必要な時点での全体構造の状態を構成する部品のバージョンが選ばれる。なお、"Goldstein, 'Layered Networks as a Tool for Software Development', proceedings of 7th International Conference on Artificial Intelligence, pp. 913-919, 1981"では、レイヤを用いた方法をソフトウェアのバージョン管理に利用した例を示している。

【0012】また、オブジェクト指向データベースのバージョン管理について記述した"Cellary, 'Consistency of Versions in Object-Oriented Databases', proceedings of International VLDB Conference, pp. 432-441, 1990"では、特定の全体構造に属する部品にバージョン名を対応づけることによって、指定したバージョンの全体構造の復元を容易にする手法を示している。

また、ハイパーテキストのバージョン管理について記述した"Osterbye, 'Structural and Cognitive Problems in Providing Version Control for Hypertext', Proceedings of ACM Conference on Hypertext (ECHT'92), pp. 33-42, 1992."では、ハイパーテキストに含まれる個々のノードのバージョンから、作成日時などの条件で適切なものを選択することによって、コンフィギュレーションを構成する手法を示している。

#### 【0013】

【発明が解決しようとする課題】上記のように、従来のバージョン管理にあっては、文書の全体構造を固定化しておけば過去の文書のバージョンを復元することが可能であるが、実際の文書の編集においては、図86～図89に示したように文書の全体構造の変更も内容部の変更に伴って行われる場合が珍しくはない。また、実際の文書執筆の過程においては、下記に示す例(1)～(3)のように、編集がなされた文書に対して、編集前の文書を復元させる必要が生ずる場合が多々ある。

【0014】(1) 特定の日時における文書の取り出し  
例えば、既に削除してしまった過去に書いた文書を取り出すために、日時を指定して、過去の時点で編集してい

た文書のバージョンを復元させる。

(2) 特定の目的で使用した文書の取り出し  
例えば、中間報告等、これまでの執筆過程で一時的にまとめて配布した文書のバージョンを復元させる。

(3) 特定の人に更新された文書の取り出し  
例えば、複数の担当者で執筆がなされているときに内容をチェックするために、特定の人による更新が行われる前のバージョンや、行われた後のバージョンを復元させる。

【0015】また、過去に存在したバージョンを復元することの必要性の他に、過去に存在しなかった文書の構造を取り出すことが必要になることがある。一般に仕事かどのように進行するかは予め決められないことが多く、進行に伴って予測していなかった様々な要求が現れるからである。そのため、あらかじめ担当者とマージの手順をきちんと決めて文書の執筆を進めても、途中でうまくゆかなることが多い。要求を満たすためには、それまでに作ってきた文書を当初想定していなかった状況に合わせて再構成する必要が生ずる。例えば、複数人で一つの調査報告書を書いているときに、現在執筆中の最新報告書の章立てではなく過去に別の目的で作った報告書の章立てに、他のメンバーの書いた最新の情報を取り入れて文書を作る必要が生じた場合、取り出されなければならない文書構造は、過去に存在したものではなく新しい目的のために過去の部品から合成したものになる。

【0016】しかしながら、従来のバージョン管理にあっては、文書の全体構造を固定化しておくことを前提として過去の文書の復元が可能となるものであるため、上記のような実際の文書処理作業に即した復元処理を行うことができなかった。なお、内容部及び文書の全体構造の変更を自由に行った場合であっても、文書の全体構造と内容部との両方のバージョンを指定するようにすれば、過去の文書構造を復元を実現することが可能であるが、このためには、利用者は文書の全体構造の変更過程の全容を把握しておかなければならないため、利用者に非常に困難且つ煩雑な作業を強いることになって実用的ではない。

【0017】更に言えば、実際の文書執筆の過程では、自分の担当している内容部の変更の影響が他の内容部の変更につながることも多いので、自分の担当範囲以外の内容部についても一貫して変更できる方が望ましい。すなわち、他の担当者によって編集集中の内容部を同時に編集することや、他の担当者に断りなく文書の全体構造を変更すること、実際の文書執筆の過程では必要となる。このように各担当者が自由に編集等を行った場合にあっては、各担当者毎の過去の文書を復元できれば支障はないが、従来のバージョン管理にあっては、文書の全体構造を固定化しておくことを前提として過去の文書の復元が可能となるものであるため、各担当者は、常に編集上の制約を守らなければならず、執筆作業

を煩わしいものにしていた。

【0018】なお、上記した多重文書処理システム（特開平7-44563号公報、特開平6-131343号公報）にあっては、ノードに付与された属性を指定することにより担当者毎の過去の文書を復元することも可能であるが、いずれのノードをどのように多重化し、また、これらのノードをどのような区別の仕方をするかについては、全て利用者に任されているため、利用者は極めて煩雑な操作を強いられることとなる。更に言えば、共同執筆や或る時点でのバージョンの復元に当該多重文書処理システムを利用することを考えると、複数の担当者が作成した部分的に内容が異なる複数の文書を管理する能力を持っているが、実際に共同執筆や必要なバージョンの復元を可能にするためには、ノードの多重化のタイミングやそのときに与える属性について煩雑な取り決めが必要になる。したがって、実際の利用に際しては極めて煩雑な操作や知識が必要とされ、当該多重文書処理システムは実質的にはバージョンの復元や共同執筆を支援するための手段とは言えなかった。

【0019】また、従来のバージョン管理技術は、本質的に過去に存在していた構造の中から必要なものを選択する技術であるため、利用者の指示にしたがって当初想定していなかった新しい状況にに応じた適切な構造を取り出すことはできなかった。例えば、導出木を用いる方法では、取り出せる構造は導出木のノードとして表される過去の状態だけであり、また、“Cellary”の方法では、過去に存在したバージョンに名前を付けるので過去に存在した構造しか取り出せない。

また、レイヤによる方法は、レイヤの順序を変えることで過去になかった構造も作り出せるが、取り出したい構造が取り出せるか、また、取り出すためにどのようにレイヤの順序を決めればよいのかを知ることができない。したがって、新しい状況に対応するための適切な構造を利用者が指示して取り出すことはできない。また、“Osterbye”の方法も、適切な条件を与えれば必要な構造を取り出せる可能性を示しているが、実際にどのような条件を与えればよいのかは示していない。したがって、新しい状況に対応するための適切な構造を利用者が指示して取り出すことはできない。

【0020】本発明は、上記従来の事情に鑑みなされたもので、文書データ等のグラフ構造をなす構造化データについて、利用者に自由に編集をさせても、過去のデータ構造の復元や新しい状況に合わせた適切なデータ構造の作成を可能ならしめることを目的とする。更に具体的に言えば、本発明は、執筆者に編集上の制約を与えずに文書を自由に編集させても、過去の文書の復元や共同執筆を支障無く実現させることを目的とする。また、本発明は、文書の全体構造が変更された場合にあっては、必要な過去の文書バージョンを支障無く復元できる構造化文書を構成することを目的とする。また、本発明は、利

用者による簡易な指定操作によって、適切な過去の文書バージョンを復元させることを目的とする。また、本発明は、利用者による簡易な指定操作によって、過去に作ってきた文書を元にして、当初想定していなかった新しい状況に合わせた適切な文書を作り出すことを目的とする。

【0021】

【課題を解決するための手段】本発明の構造化データ処理装置では、ノード保持手段にグラフ構造を成している構造化データのノード情報を保持し、リンク保持手段に構造化データのノード間の関連情報を表すリンク情報を保持し、リンク間関係保持手段に同一のノードを始点とする複数のリンクの内の同一の編集処理に係るリンクを集合として保持するとともに各集合中の含まれるリンク間の位置関係情報を保持している。そして、編集対象となる構造化データは、リンク関係保持手段に保持されたリンク集合中のいずれか1つの集合を特定することにより、ノード保持手段及びリンク保持手段に保持された情報に基づいて特定されてカレント構造管理手段で管理されており、この特定された構造化データを編集手段により編集処理を行う。そして、編集処理によりリンクに変更が加えられたときには、登録手段が変更されたリンクを含むリンク集合を生成してリンク間関係保持手段に保持させる。

【0022】なお、前記編集手段はノードを追加する処理も行い、また、前記登録手段は追加されたノードをノード保持手段に保持させる処理も行う。また、編集処理に係る情報として選択属性を編集処理により変更されたリンク集合に付与する選択属性対応付け手段を更に備えており、特に選択属性としてバージョン情報を付与するときには、選択属性対応付け手段は編集処理に係る構造化データ中の他のリンクに対してもバージョン情報を付与する。

【0023】また、本発明の構造化データ処理装置では、条件入力手段から構造化データを再構成させるための選択条件を入力すると、リンク集合選択手段が入力された選択条件に対応する選択属性が付与されているリンク情報をリンク間関係保持手段から選択し、データ構成手段が選択されたリンク集合に基づいて構造化データを再構成する。なお、リンク集合選択手段は、入力された選択条件に対応するリンク集合が複数存在する場合には、利用者情報を優先させる、日時情報を優先させる、バージョン情報を優先させる等といった、予め設定された選択規則に従っていずれか1つのリンク集合を選択し、データ構成手段は当該リンク集合に基づいて構造化データを再構成する。

【0024】ここで、本発明を木構造を成している構造化文書データの処理に適用した場合には、本発明は下記のような文書処理装置又は文書処理方法として実現される。なお、以下の記載において、論理単位はノードとリ

ンクとの少なくともいずれか一方を含むデータ構造中の単位(部分データ)に該当する。

【0025】本発明の文書処理装置では、論理単位保持手段に木構造を成している構造化文書の内容の一部分である内容単位又は直下の階層の構造との関連位置を表す参照単位の少なくともいずれか一方を論理単位として保持させ、論理単位グループ保持手段に構造化文書を構成する木構造において兄弟の関係にある全ての論理単位から成る論理単位のグループ(集合)を保持させる。なお、参照単位は、該参照単位の直下に位置する少なくとも1つの論理単位グループの集合を表すものであり、参照単位で示された論理単位グループによって文書全体が構成される。また、兄弟の関係とは、同じ論理単位(親)を直上にもつ論理単位同士の関係をいう。

【0026】更に、本文書処理装置では、文書編集手段によって文書構造を編集すると、論理単位グループ生成手段が当該文書編集に対応する論理単位グループを新たに生成し、参照関連付手段が新たに生成された論理単位グループと所定の参照単位を関連付ける。この結果、新たに生成された論理単位グループは既存の構造化文書中に組み入れられ、文書編集に応じた構造化文書が作成される。また、新たに生成された論理単位グループは、対応する編集前の論理単位グループを同一の集合に追加されて構造化文書中に組み入れられ、これら論理単位グループは同じ階層の集合にまとめられて論理単位グループ保持手段に保持される。

【0027】また、本発明の文書処理装置では、更に、選択情報対応付け手段によって論理単位グループと該論理単位グループを選択するための選択情報とを対応付け、条件指定手段によって文書構造を構成するために前記選択情報を含む選択条件を指定すると、選択情報対応付け手段によって対応付けられている選択情報に基づいて、論理単位グループ選択手段が論理単位グループ保持手段に保持されている論理単位グループを選択する。そして、文書構成手段が論理単位グループ選択手段により選択された論理単位グループに基づいて文書を構成する。

【0028】この選択情報としては、例えば、文書のバージョン情報、文書の更新日時、文書の利用者情報等が用いられ、これら選択情報が選択情報対応付け手段によって論理単位グループ保持手段に保持されている論理単位グループに対応付けられる。したがって、選択情報に基づいて構成された文書は指定された文書のバージョン情報、文書の更新日時、文書の利用者情報等に応じたものであり、過去の文書バージョンがこれら選択情報に基づいて復元される。

【0029】また、本発明の文書処理装置では、更に、選択情報対応付け手段により論理単位グループに対応付けられる選択情報を、バージョン情報、更新日時情報、利用者情報とし、条件指定手段により指定される選択条

件は文書のバージョンと日時情報と利用者情報を含むものとする、そして、論理単位グループ選択手段に備えられた検索手段によって、条件指定手段により指定されたバージョン情報に一致する論理単位グループに対応付けられた更新日時情報以降で、指定された日時情報以前の更新日時情報と対応付けられ、且つ、指定された利用者情報に一致する論理単位グループを検索し、更に、論理単位グループ選択手段に備えられた選択手段によって、検索手段による検索結果で複数の論理単位グループが存在した場合には、いずれか1つの論理単位グループを選択する。そして、この選択された論理単位グループを表示手段が表示し、利用者に対応し且つ指定された日時を基準とした最新のバージョンの文書が検索されて表示される。

【0030】また、本発明の文書処理方法では、構造化文書の内容の一部分である内容単位又は下位階層の構造を表す参照単位の中から編集対象を指定し、この指定した内容単位又は参照単位を編集すると、編集に対応して、構造化文書を構成する木構造において兄弟の関係にある全ての内容単位あるいは参照単位から成る論理単位グループが生成される。そして、新たに生成された論理単位グループを選択するための選択情報が入力されると、入力された選択情報と新たに生成された論理単位グループとを関連付け、該論理単位グループを前記指定した内容単位あるいは参照単位を含む論理単位グループの集合に追加する。したがって、新たに生成された論理単位グループは既存の構造化文書中に組み入れられ、文書編集に応じた構造化文書が作成される。

【0031】また、本発明の文書処理方法では、文書の内容の一部分である内容単位と下位階層の構造を表す参照単位を有する構造化文書を選択するための選択条件を入力すると、構造化文書を構成する木構造において兄弟の関係にある全ての内容単位又は参照単位からなる論理単位グループの中から、入力された選択条件と論理単位グループに対応付けられている選択情報とに基づいて論理単位グループが選択され、選択された論理単位グループより文書が構成されて表示される。したがって、バージョン情報、更新日時情報、利用者情報等といった選択情報を含む選択条件を入力することによって、当該選択情報に対応する文書が構成されて表示される。

【0032】すなわち、本発明によれば、文書の全体構造と部分構造について区別なく適用できるバージョン管理を可能にし、文書の内容及び構造に自由に変更を加えても、必要なバージョンの文書を取り出すことが可能となっている。換言すれば、本発明では、文書の構造を、全体構造と部分構造のバージョンを同等に管理できるデータ構造として、文書データとしては文書の構成要素の全てのバージョンを含んだものとして、この中から適切な抽出条件によって抽出した部分構造を組み合わせ文書全体を構成する。



【0033】また、本発明によれば、共同執筆において、それぞれの担当者が与えた条件に適合するバージョンの文書を一貫して編集することが可能化し、各担当者が混乱することなく編集処理を行うことが可能になる。すなわち、本発明では、バージョン指定情報と、利用者指定情報と、日時指定情報とを組み合わせ、共同執筆者が編集の対象とするべきバージョンを取り出すための条件を定義してあり、各担当者の執筆作業の際に、対象としている文書部分にこれらの情報を関連付けておくことにより、文書を自由に編集しても、これら情報に基づいて文書を構成することにより各執筆者にとって編集すべきバージョンの文書が復元される。

#### 【0034】

【発明の実施の形態】本発明の一実施例に係る文書処理装置及び方法を説明するに当たって、まず、本実施例における構造化文書の構成及びバージョン管理を図3～図8を参照して説明する。なお、本実施例においては、請求項に記載した、木構造を成している構造化文書の一部である内容単位は内容部として、構造化文書の下層の構造部分との関連位置を示す参照単位はアンカーとして、実現され、更に、これら内容部とアンカーとの少なくともいずれか一方を論理単位とし、木構造の同じ階層に位置する論理単位から成る論理単位グループはストリームとして、論理単位グループを選択するための選択情報はストリーム属性として実現される。

【0035】本実施例では、ストリームに基づく木構造の構造化文書を利用しており、ストリーム単位でバージョンの管理を行うことにより文書の全体構造と部分構造を統一的に扱えるようにして、文書の全体構造と内容部とのバージョンを同等の方式によって管理することを可能にしている。すなわち、本実施例では、文書内容の変更、文書内容や文書の部分構造の挿入、文書内容や文書の部分構造の削除、文書内容や文書の部分構造の順序の変更、文書内容や文書の部分構造の移動、等といった一般的な文書構造の編集に係る基本操作と、適切な内容部やストリームの生成の操作を組み合わせることによって、任意の文書構造の生成、変更、復元等が可能になっている。

【0036】図3に示す文書構造の例では、ノードAを文書ルートのアンカーとすると、例えば、アンカーB及び内容部(u)を含むストリームaが章、アンカーC及びDを含むストリームbが節、内容部(v)と(x)を含むそれぞれのストリームc、dが項に該当することとなる。そして、新たな内容部を作成して既存の内容部と入れ替えるといった編集処理を行う場合には、例えば図4～図8に示すようなストリーム単位での処理を行って文書の内容部及び全体構造を変更する。

【0037】まず、図3に示した元の文書に対して内容部を変更する場合には、図4に示すように、新たな内容部(y)を含むストリームd2を作り、既存の内容部

(x)を含むストリームd1と新たなストリームd2とを入れ替えるとともに、古いストリームd1と新しいストリームd2とをアンカーDが参照する集合として記憶管理する。

【0038】また、図3に示した元の文書に対して新たな部分構造を挿入する場合には、図5に示すように、アンカーC及びDを含む既存のストリームb1に対して更にアンカーEを含む新たなストリームb2を作るとともに、内容部(z)を含んだ新たなストリームeを作り、新たなストリームb2を既存のストリームb1と入れ替えるとともに、新たなストリームb2の各アンカーC、D、Eによってそれぞれストリームc、d、eを参照させる。なお、この場合にあっても、同一のアンカーBで参照される同じ階層のストリームb1、b2はアンカーBが参照する集合として記憶管理する。

【0039】また、図3に示した元の文書に対して既存の部分構造Dを削除する場合には、図6に示すように、アンカーC及びDを含む既存のストリームb1に対してアンカーDを削除した新たなストリームb3を作り、既存のストリームb1と新たなストリームd3とを入れ替えるとともに、古いストリームb1と新しいストリームb3とをアンカーBが参照する集合として記憶管理する。また、新たなストリームb3のアンカーCでストリームcを参照させるとともに、ストリームdを古いストリームb1のアンカーDによって参照されるストリームとして記憶管理する。

【0040】また、図3に示した元の文書に対して部分構造C、Dの順序を入れ替える場合には、図7に示すように、アンカーC及びDを含む既存のストリームb1に対してアンカーCとDの順序を入れ替えた新たなストリームb4を作り、既存のストリームb1と新たなストリームd4とを入れ替えるとともに、古いストリームb1と新しいストリームb4とをアンカーBが参照する集合として記憶管理する。

【0041】また、図3に示した元の文書に対して部分構造Dを移動させる場合には、図8に示すように、アンカーC及びDを含む既存のストリームb1に対してアンカーDを削除した新たなストリームb3を作り、既存のストリームb1と新たなストリームd3とを入れ替えるとともに、内容部(u)及びアンカーBを含む既存のストリームa1に対して更にアンカーDを含ませた新たなストリームa2を作り、既存のストリームa1と新たなストリームa2とを入れ替え、アンカーDにはそのままストリームdを参照させる。なお、この場合にあっても、同一のアンカーAで参照される全てのストリームa1、a2はアンカーAが参照する集合として記憶管理し、同一のアンカーBで参照される全てのストリームb1、b3はアンカーBが参照する集合として記憶管理する。

【0042】上記のように、編集処理されたストリーム

は編集前のストリームと同一の集合として記憶され、これら集合化されたストリームは同一のアンカーによって参照される。したがって、後述するように、アンカーで示される集合の内から所定の条件に適合するストリームを選択し、これらストリームによって文書を構成することにより、編集処理によって文書の全体構成を変更した場合にあっても、編集前の過去の文書を復元することができる。

【0043】このような文書処理を実行する本実施例の文書処理装置は、図1に示す構成を有している。半導体メモリや磁気記憶装置等から構成されたメモリ1には、論理単位保持手段2、論理単位グループ保持手段3、選択情報対応付け手段4が構成されており、これら手段2～4によって構造化文書に関する情報を記憶管理する。論理単位保持手段2はアンカー又は内容部の少なくともいずれか一方を含む論理単位を保持し、論理単位グループ保持手段3は論理単位から成るストリームを保持する。

【0044】選択情報対応付け手段4は、論理単位グループ保持手段3に保持されているストリームとストリームを選択するためのストリーム属性とを対応付けており、これによって、ストリーム属性を指定することにより、論理単位グループ保持手段3から対応するストリームを選択することが可能となっている。したがって、構造化文書は図2に示すように各ストリーム毎にストリーム属性を対応付けたデータ構造としてメモリ1に格納されており、本実施例における以下の説明では、これらストリームとストリーム属性との組をストリームセットとして定義する。また、ストリームセットは、同一のアンカーで参照され且つ文書構造中の同じ階層に位置するストリームとストリーム属性との組を複数個包含する場合もあり、この場合にはストリームセットは複数のストリームを包含した集合となる。また、文書構造の最上位のノードはルートアンカーであり、このルートアンカーから下位のストリームセットへ、当該ストリームセットに含まれているアンカーから更に下位のストリームセットへといったように、これらストリームセットはアンカーによる参照によって木構造化されている。

【0045】文書編集手段5は、キーボードやマウス等といった入力手段からの利用者による指示に基づいて、メモリ1に保持されている構造化文書に編集処理を施し、図4～図8に示したような内容部や部分構造（ストリーム）に対する変更処理を行う。論理単位グループ生成手段6は、文書編集手段5によって編集処理がなされたことに基づいて対応する新たなストリームを生成する処理を行い、例えば図4に示したように、内容部（x）を内容部（y）に変更した場合には、内容部（y）を含む新たなストリームd2を生成する。

【0046】参照関連付け手段7は論理単位グループ生成手段6によって新たに生成されたストリームを編集対

象となった元のストリームと同一の集合に追加して論理単位グループ保持手段3に保持させる処理を行い、例えば図4に示したように、既存のストリームd1に対して新たなストリームd2が生成された場合には、当該ストリームd2をストリームd1と同じ集合（ストリームセット）に追加する。

【0047】条件指定手段8は、利用者が操作するキーボードやマウス等の入力手段を有し、ストリーム属性を含む選択条件が入力されるとこれを論理単位グループ選択手段9へ出力する。なお、本実施例では文書のバージョン情報、文書の更新日時情報、文書の利用者情報等がストリーム属性として設定され、このようなストリーム属性を含む選択条件が論理単位グループ選択手段9に入力される。論理単位グループ選択手段9は、条件指定手段8から入力された選択条件に基づいて論理単位グループ保持手段3から選択条件に適合するストリームを選択する処理を行い、この選択処理を行うために検索手段10と選択手段11とを有している。

【0048】検索手段10は論理単位グループ保持手段3に保持されているストリーム中に選択条件に適合するストリームが存在するか否かを検索処理し、選択手段11はこの検索によって適合するストリームが存在するときには当該ストリームを選択する処理を行う。例えば、選択条件として文書のバージョンと日時情報と利用者情報が指定された場合、検索手段10によって、バージョン情報に対応する更新日時情報以降で、指定された日時情報以前の更新日時情報と対応付けられ、且つ、指定された利用者情報に一致するストリームを検索し、更に、選択手段11によって、この選択条件に適合するストリームを選択する。なお、選択手段11は検索手段10による検索によって、上記のように選択条件に含まれる全ての属性に適合するストリームが存在しない場合には、この属性の内からバージョン情報に一致するストリームを選択する。

【0049】文書構成手段12は論理単位グループ選択手段9によって選択されたストリームに基づいて文書を構成する処理を行い、文書構造表示手段13は文書構成手段12によって構成された文書をディスプレイ14に表示させる処理を行う。未表示情報表示手段15は、文書構造表示手段13によって文書の構成部分として表示されていない未表示のストリームをマークとしてディスプレイ14に表示させる処理を行う。すなわち、論理単位グループ選択手段9によって選択されたストリームを含むストリームセットが他のストリームを含んでいる場合には、当該他のストリームを後述（図17）するようにマーク化した候補として、文書構成手段12によって構成された文書とともにディスプレイ14に表示する。

【0050】論理単位グループ指定手段16は、キーボードやマウス等の入力手段を有し、利用者からの入力に基づいて未表示情報表示手段15によって表示されたマー

力を指定する処理を行う。論理単位グループ再選択手段17は、論理単位グループ指定手段16によってマークを指定したことに基づいて、論理単位グループ選択手段9に、当該指定されたマークに対応するストリームと論理単位グループ選択手段9で既に選択されているストリームとを入れ替える処理を行わせる。すなわち、ディスプレイ14に或るストリームのコンテンツ部(テキスト)が文書の構成部分として表示されている場合、マークを選択することによって、他のストリームのコンテンツ部(テキスト)が入れ替えて表示される。また、入れ替えられたストリーム間でアンカーの数や位置等といったアンカーに係る構成が異なる場合には、表示されている文書の全体構成も変更される。

【0051】ここで、上記した論理単位グループ選択手段9が行う選択条件に基づくストリームの選択処理を図9を参照して更に詳しく説明する。なお、図9に示す例では、各ストリームの属性として、“Usui”、“Sato”、“Tani”といった利用者名からなる利用者情報と、更新日時情報と、バージョン情報とが対応付けられており、入力される選択条件には、文書を参照して処理を行うための基準日時(基準日時情報)と、基準となるバージョン情報(基準バージョン情報)と、利用者情報と、が含まれている。また、縦軸の日時に沿って利用者情報毎のストリーム(1)～(5)を示してある。

【0052】この例では、利用者の“Usui”さんが参照日時として図中に点線で示した時点で実際に文書を参照する場合を考える。まず、選択条件として何も指定しない場合には、参照日時を基準として最新のストリーム(5)が選択される。また、“本人”(すなわち、“Usui”)とした利用者情報のみを指定した場合には、ストリーム属性が“Usui”である最新のストリーム(1)が選択される。また、“Sato”とした利用者情報のみを指定した場合には、ストリーム属性が“Sato”である最新のストリーム(5)が選択される。また、図中に示す基準日時のみを指定した場合には、ストリーム属性の更新日時が基準日時までの最新のストリーム(4)が選択される。

【0053】また、基準日時として図示された日時を指定し、且つ、利用者情報として“Sato”を指定した場合には、ストリーム属性が“Sato”で且つ更新日時が基準日時までの最新のストリーム(3)が選択される。また、図示された基準バージョンのみを指定した場合には、基準バージョン以降の最新のバージョン(5)が選択される。また、図示された基準バージョンと基準日時として図示された日時を指定した場合には、基準バージョン以降で、基準日時までの最新のストリーム(4)が選択される。

【0054】また、図示された基準バージョンと基準日時として“最初”を指定した場合には、基準バージョン

以降で、最初のストリーム(2)が選択される。また、図示された基準バージョンと、利用者情報として“本人”を指定した場合には、基準バージョン以降にはストリーム属性が“Usui”であるストリームがないため、基準バージョンのストリーム(2)が選択される。このように、ストリーム属性に基づいて、条件指定手段8から指定された選択条件に適合するストリームが選択され、当該ストリームに基づいて文書構成手段12によって文書が構成される。

【0055】次に、上記構成の文書処理装置によって実施される文書編集処理の概要を図10～図36に示す具体例を用いて説明する。なお、以下の説明によって、上記した各構成手段による機能も更に明確化されることとなる。この具体例では、図10に示す編集課程によってSatoさんとTaniさんが「発明説明書」という名前の文書を共同で執筆している。なお、個々の編集作業の開始から終了までをセッションと称し、開始は文書編集ツールで文書を開くことによって行われ、終了は文書を閉じることによって行われる。

【0056】図10では、横軸に時間を取り、1～7の各セッションは担当者“Sato”、“Tani”に対応する水平方向の線分で示される。また、各線分の後端の黒い三角形は文書の保存が行われた時点を示しており、また、保存のマークからの垂直方向の線でセッションの名前を定義したことを示してある。例えば、セッション4では、1995年1月4日に文書を保存し、この際に「中間報告」という名前のバージョンを定義していることを示している。なお、この具体例では、それぞれのセッションは、時間が重ならないようにしているが、これは説明を無用に複雑にしないためであり、本発明はセッションの時間が重なっても何らの支障を生じるものではない。

【0057】利用者Satoさんが新しく文書「発明説明書」を作成して保存するセッション1について、図11～図15を参照して説明する。まず、図11に示すように、1995年1月1日の15:00に、Satoさんが文書編集手段5の文書エディタを起動すると、ディスプレイ14に図に示す対話ツールが表示されて新規文書名の入力を利用者に促す。これに対して、Satoさんが文書ルート名として「発明説明書」を入力し、設定完了を指示する「ok」ボタンをクリックすると、文書の抽出を開始するルートアンカーの名前が設定される。なお、このルートアンカーは文書を構成するストリームセットを参照するものである。

【0058】次いで、図12に示すように、文書エディタが抽出した文書構造をディスプレイ14に表示する。この文書構造の表示は、アンカーの位置を示す白い三角形のマークと、アンカー名と、アンカーの参照するストリームのコンテンツ部とを表示することにより行われる。なお、アンカーの位置を示すマークのインデントの深さ



は、文書構造中におけるストリームの階層の深さを示しており、下層のストリームを参照するアンカーになる程、マーカの表示される位置は図中の右方向となる。

【0059】すなわち、「発明説明書」という名称のルートアンカーで参照されるストリームに未だ内容部（テキスト内容）が未だ含まれていないときには、図12（1）に示すように、ルートアンカー名称「発明説明書」が表示され、当該ルートアンカーが参照する空のストリームが図中に点線で示す領域として表示される。そして、Sat oさんが編集作業を行って、ルートアンカーが参照するストリームに内容部（u）としてテキストを記述すると、このテキスト内容が図12（2）に示すように上記の領域内に表示される。この編集の結果、論理単位グループ生成手段6によって、図13に示すように、新しく作成された内容部（u）だけからなるストリームaが生成され、このストリームaがルートアンカーAによって参照された文書構造が構成される。

【0060】次いで、編集処理を終了して、1995年1月1日の20:00に、Sat oさんが図12に示す文書エディタの「save」ボタンをクリックし、文書を保存する指示を入力すると、ディスプレイ14には図14に示す対話ツールが表示されて保存時の指定を促す。これに対して、この例では、Sat oさんは何も設定することなく、「ok」ボタンをクリックして作成した文書の保存処理を指示する。なお、この対話ツールには文書ルートとしてルートアンカー名「発明説明書」が表示されている。

【0061】この保存処理の結果、メモリ1には、図15に示す文書構造が格納される。すなわち、この文書構造には、ルートアンカーA、ルートアンカーAで参照されるストリームセットa、ストリームセットに含まれるストリームa1、ストリームa1に含まれる内容部（u）が新しく生成され、ストリームa1にはストリーム属性として、編集した利用者の情報である“Sat o”、データを保存（更新）した日時である「1995.1.1 20:00」が選択情報対応付け手段4によって対応付けられる。

【0062】次に、図10に示したセッション2においては、Sat oさんが自分がセッション1で作成した文書「発明説明書」に編集を加え、編集結果をメモリ1に保存する。まず、図16（1）に示すようにディスプレイ14には文書フォルダが表示されており、1995年1月2日の15:00に、Sat oさんが文書フォルダの「発明説明書」アイコンをダブルクリックして選択すると、文書編集手段5の文書エディタが起動される。この結果、ディスプレイ14には図16（2）に示す抽出条件設定ツールが表示され、抽出条件の設定を促す。これに対して、Sat oさんが条件指定手段8から利用者情報として本人を表す変数「%本人」を入力し、設定完了を指示する「ok」ボタンをクリックすると、編集処

理が開始される。なお、「%本人」「%最初」等の「%」記号は、文書抽出を開始する時に、システムが値を計算して設定する変数であることを示している。

【0063】上記の設定は、本人が更新した最新の構成部分（ストリーム）を含む文書を抽出して編集対象とすることを示しており、図17（1）に示すように、文書エディタが文書エディタは、セッション1で保存したときの文書構造をメモリ1から抽出してディスプレイ14に表示する。次いで、Sat oさんが文書の編集作業として、図17（1）に示すようにルートアンカーである「発明説明書」が参照するストリームに、アンカーとして「請求項」を追加し、更に、アンカー「請求項」が参照するストリームに内容部（v）としてテキストを記述すると、このテキスト内容が図17（2）に示すようにアンカー「請求項」が参照する領域内に表示される。

【0064】この編集の結果、図18に示すように、論理単位グループ生成手段6によって、ルートアンカーAによって参照されて内容部（u）とアンカーBとを含む新たなストリームa2が生成されるとともに、アンカーBによって参照されて内容部（v）を含む新たなストリームb1が生成される。なお、図17に示すように、文書エディタ中には未表示情報表示手段15によって候補マーカMaが表示され、アンカーAが参照するストリームセットにはディスプレイ表示されているストリームの他に、他のストリームが候補として存在することを示している。この場合、アンカーAが参照するストリームの候補にはa1とa2とがあり、一方のストリームについてディスプレイ表示されているときには、他方のストリームはマーカMaによってディスプレイ表示される。

【0065】次いで、編集処理を終了して、1995年1月2日の20:00に、Sat oさんが図17に示す文書エディタの「save」ボタンをクリックし、文書を保存する指示を入力すると、ディスプレイ14には図14に示したと同様な対話ツールが表示されて保存時の指定を促す。これに対して、Sat oさんがセッション1のときと同様にして文書の保存処理を指示すると、メモリ1には、図19に示す文書構造が格納される。

【0066】すなわち、この文書構造には、ルートアンカーA、ルートアンカーAで参照されるストリームセットa、ストリームセットaに含まれるストリームa1及びa2、アンカーBによって参照されるストリームセットb、ストリームセットbに含まれるストリームb1、が含まれ、更に、ストリームa1には利用者情報の“Sat o”、データ保存（更新）日時情報の「1995.1.1 20:00」といったストリーム属性、ストリームa2には利用者情報の“Sat o”、データ保存（更新）日時情報の「1995.1.2 20:00」といったストリーム属性、ストリームb1には利用者情報の“Sat o”、データ保存（更新）日時情報の「1995.1.2 20:00」といったストリーム属

性、が選択情報対応付け手段4によって対応付けられる。

【0067】次に、図10に示したセッション3においては、Satoさんが自分がセッション2で編集した文書「発明説明書」に更に編集を加え、編集結果をメモリ1に保存する。まず、1995年1月3日の15:00に、セッション2と同様にしてSatoさんが文書「発明説明書」の編集を開始すると、文書エディタはセッション2で保存したときの文書構造をメモリ1から抽出して図20(1)に示すようにディスプレイ14に表示する。

【0068】これに対して、Satoさんが、文書の編集作業として、図20(1)に示すように請求項の欄に項目2を追加すると、論理単位グループ生成手段6が請求項のアンカーBが参照するストリームとして、項目1のアンカーCと項目2のアンカーDを含むストリームを作成する。そして、図20(2)に示すように、この項目2にテキストを記述すると、このテキストからなる内容部(x)がアンカーDによって参照される新たなストリームとして論理単位グループ生成手段6によって生成される。

【0069】したがって、文書エディタ内には図21に示すように、ルートアンカーAによって参照されて内容部(u)とアンカーBとを含むストリームa2と、アンカーBによって参照されてアンカーCとDとを含むストリームb2と、アンカーCによって参照されて内容部(v)を含むストリームc1と、アンカーDによって参照されて内容部(x)を含むストリームd1と、が生成される。なお、上記の編集処理によってアンカーBが参照するストリームが複数の候補となるので、図20に示すように、これらストリームb1、b2の内の未表示のストリームを示すマークMbがディスプレイ表示される。

【0070】次いで、編集処理を終了して、1995年1月3日の20:00に、Satoさんがセッション2と同様にして文書を保存する指示を入力すると、メモリ1には図22に示す文書構造が格納される。すなわち、この文書構造には、図19に示した構造に対して、ストリームセットbにアンカーCとDを含むストリームb2が追加され、更に、アンカーCで参照されるストリームセットc、ストリームセットcに含まれる内容部(v)を有したストリームc1、アンカーDで参照されるストリームセットd、ストリームセットdに含まれる内容部(x)を有したストリームd1、が追加される。更に、この文書構造の新たに追加されたストリームb2、c1、d1に対しては、利用者情報の「Sato」、データ保存(更新)日時情報の「1995.1.320:00」といったストリーム属性が選択情報対応付け手段4によってそれぞれ対応付けられる。

【0071】次に、図10に示したセッション4におい

ては、Satoさんが作成した文書「発明説明書」にTaniさんが編集を加え、その編集結果を「中間報告」バージョンとして保存する。まず、1995年1月4日の10:00にTaniさんがセッション2と同様にして編集を開始すると、図23に示すように抽出条件設定ツールがディスプレイ14に表示される。これに対して、Taniさんは設定条件として何も指定せずに「ok」ボタンをクリックして、図24に示すような文書エディタをディスプレイ表示させて編集を実行する。なお、このように何も条件を設定せずに編集を開始すると、利用者に関わらず最新の文書がメモリ1から抽出されて編集対象となる。また、ここで、Taniさんはこれまでに、文書の更新を行っていないので、Satoさんのように利用者条件として条件指定手段8から「%本人」を指示すると、文書として何も抽出されないことになる。

【0072】図24に示すように、文書エディタはセッション3で保存したときの文書構造を抽出してディスプレイ表示しており、Taniさんが編集作業として、図に示すように請求項のアンカーBが参照するストリームとして、項目1と項目2の間に項目1.5を追加する。この編集の結果、図25に示すように、論理単位グループ生成手段6によって、アンカーBによって参照されてアンカーC、D、Eを含む新たなストリームb3、アンカーEによって参照される項目1.5の内容部(z)を含む新たなストリームe1が生成される。

【0073】次いで、編集処理を終了して、1995年1月4日の15:00に、Taniさんが文書エディタの「save」ボタンをクリックし、文書を保存する指示を入力すると、ディスプレイ14には図26に示す対話ツールが表示されて保存時の指定を促す。これに対して、Taniさんはバージョン情報として「中間報告」を設定して、「ok」ボタンをクリックして作成した文書の保存処理を指示する。

【0074】この保存処理の結果、メモリ1には、図27に示す文書構造が格納される。すなわち、図22に示したセッション3での文書構造に対して、アンカーBで参照されるストリームセットbにアンカーC、D、Eを含む新たなストリームb3が追加され、更に、アンカーEによって参照されるストリームセットe、及び、ストリームセットeに含まれる新たなストリームe1、が追加される。なお、この編集処理によって生成されたストリームb3、e1には、ストリーム属性として、編集した利用者の情報である「Tani」、データを保存(更新)した日時である「1995.1.415:00」、バージョン情報である「中間報告」が選択情報対応付け手段4によって対応付けられる。また、この編集処理に関わったストリームa2、c1、d1のストリーム属性にも、選択情報対応付け手段4によってバージョン情報の「中間報告」が追加して対応付けられる。

【0075】次に、図10に示したセッション5においては、Satoさんが自分が編集した文書「発明説明書」に更に編集を加え、編集結果を保存する。なお、この編集開始時にはセッション4でTaniさんによる変更が加えられているが、抽出条件の設定により、自分が編集して保存した状態の文書について編集を再開する。すなわち、セッション2と同様にして1995年1月4日の15:00に文書エディタ起動する。そしてこの際に、条件指定手段8から入力する抽出条件設定ツールの指定を「%本人」とすることによって、論理単位グループ選択手段9は請求項のアンカーBが参照するストリームとして、Taniさんが作成したb3ではなく、Satoさんの作成したb2を選択し、セッション3でSatoさんが保存した状態の文書構造を図28(1)に示すように文書エディタで表示する。

【0076】なお、この場合の候補マーカMbは反転表示され、これによって、請求項のアンカーBが現在参照しているストリームの候補により新しいストリームがあることを示す。そして、Satoさんが編集作業として、図28(2)に示すように、項目2の内容部(x)を内容部(y)に書き換えると、図29に示すように、論理単位グループ生成手段6が項目2ののアンカーDが参照するストリームとして、内容部(x)を内容部(y)に書き換えたストリームd2を新たに生成する。

【0077】次いで、1995年1月4日の20:00にSatoさんがセッション1と同様にして文書を保存すると、メモリ1には図30に示す文書構造が格納される。すなわち、図27に示したセッション4での文書構造に対して、アンカーDで参照されるストリームセットdに内容部(y)を含む新たなストリームd2が追加され、更に、このストリームd2にはストリーム属性として、編集した利用者の情報である“Sato”、データを保存(更新)した日時である「1995.1.4 20:00」、が選択情報対応付け手段4によって対応付けられる。

【0078】次に、図10に示したセッション6においては、Taniさんが自分が編集を加えて保存した文書「発明説明書」の「中間報告」バージョンと、その後、自分以外の執筆者によって加えられた編集の結果を参照する。1995年1月5日の10:00に、Taniさんが文書「発明説明書」の編集を開始し、セッション2と同様にして、文書フォルダのアイコンを選択してダブルクリックすることで、文書エディタを起動する。この時、図31(1)に示す抽出条件設定ツールによって、文書の抽出条件の設定が促され、これに対して、Taniさんがバージョン情報として「中間報告」、利用者情報として「%本人」を指定する。この指定によって「中間報告」として指定したバージョンに、Taniさんにより更新された構成要素(ストリーム)から成る文書がメモリ1から抽出されて編集対象となる。

【0079】すなわち、論理単位グループ選択手段9が条件指定手段8から入力された条件に従って、「中間報告」を属性に含むストリームを検索し、Taniさんの作成した最新のストリームを選択する。ここで、Taniさんがセッション4でバージョンを設定後、Satoさんは編集を行っているが、Taniさんは編集を行っていないため、項目2のアンカーDが参照するストリームとして、Satoさんの作成した最新のストリームd2ではなく、「中間報告」を設定した時点でのストリームd1が選択される。この結果、図32(1)に示すように、文書エディタにはTaniさんがセッション4で保存した「中間報告」の文書構造(図24)が表示される。なお、項目2のアンカーDに対応する候補マーカMdは反転表示されて、Satoさんによって作成されたより新しいストリームの候補があることを示している。

【0080】ここで、文書エディタの「filter」ボタンをクリックすることによって、抽出条件設定ツールを再び呼び出すことができ、この呼び出したツールで条件を変更すると、新しい条件で文書構造が再抽出される。したがって、Taniさんが図31(2)に示すように、抽出条件として「%本人」を除いた指定を行うと、この指定によって、より新しいストリームが利用者の区別なく選択される。この結果、項目2のアンカーDが参照するストリームとして、Satoさんが作成した最新のストリームd2が選択され、図32(2)に示す文書が文書エディタに表示される。

【0081】したがって、図31(1)に示した最初の条件の指定では、エディタ内には図33(1)に示す文書構造が取り出され、図31(2)に示した条件の変更後は、図33(2)に示す文書構造が取り出される。すなわち、図33(1)に示す文書構造ではアンカーDが内容部(x)を含むストリームd1を参照するが、図33(2)に示す文書構造ではアンカーDが内容部(y)を含むストリームd2を参照する。なお、セッション6ではTaniさんは編集を終了するが、文書は保存せず、メモリ1に格納される文書構造は図30に示した構造のままである。

【0082】次に、図10に示したセッション7においては、Satoさんが自分が編集を行ってきた文書「発明説明書」について、自分以外の執筆者による編集の結果を反映させたものを「最終報告」バージョンとして保存する。セッション7では、1995年1月5日の15:00に、Satoさんがセッション2と同様にして文書エディタを起動させ、文書「発明説明書」の編集を開始する。

【0083】文書エディタは、図34に示すように、セッション5でSatoさんが保存したときの文書構造をメモリ1から抽出してディスプレイ表示する。この文書エディタには請求項のアンカーBが参照するストリームに、未選択のより新しい候補があることが候補マーカM

bによって示されている。この候補マーカMbをクリックすると、候補のストリームb1、b2、b3がメニューKとして表示され、このメニューKから論理単位グループ指定手段16でストリームを選択すると、論理単位グループ再選択手段17によって選択されたストリームが現在参照中のストリームと置き換えられ、更に、置き換えられたストリームより下位の構造は抽出条件に従って再構成される。

【0084】したがって、Satoさんが候補メニューKから最新のストリームb3を選択することによって、セッション5で作成した同じ文書を編集対象とできる。このセッション7の編集処理では、文書エディタ内で、図29に示したセッション5での文書構造が図33(2)に示した構造に変更される。そして、1995年1月5日の20:00に、Satoさんが文書を保存する指示を行うと、図35に示す対話ツールがディスプレイ表示されて、保存時の条件設定を促す。これに対して、Satoさんがバージョン情報として「最終報告」を指定して保存を行うと、メモリ1には図36に示すように編集処理が加えられた文書構造が格納される。すなわち、図30に示したセッション5での文書構造に対して、保存処理時に参照していたストリームa2、b3、c1、d2、e1の属性に、選択情報対応付け手段4によってバージョン情報の「最終報告」が追加して対応付けられる。

【0085】次に、文書をメモリ1から抽出してディスプレイ14に表示する処理を図37～図40に示すフローチャートを参照して更に詳しく説明する。図37に示すように、まず、利用者による条件指定手段8からの指示に従って論理単位グループ選択手段9がメモリ1からルートアンカーを選択し(ステップS1)、引き続いて、指定されたルートアンカーの下位の階層に位置する構造をメモリ1から抽出して文書をディスプレイ14に表示する処理を行う(ステップS2)。

【0086】このステップS2での下位構造抽出表示処理は図38に示す手順でなされる。まず、文書構成手段12が抽出中の文書構造にルートアンカーを組み入れ(ステップS11)、文書構造表示手段13がこのルートアンカー名を表示した文書エディタをディスプレイ14の画面に表示する(ステップS12)。次いで、ルートアンカーが参照しているストリームセットを論理単位グループ選択手段9がメモリ1から選択し(ステップS13)、このストリームセットから、利用者によって条件指定手段8から指示された条件に適合する属性のストリームを選択する(ステップS14)。

【0087】次いで、選択されたストリームを文書構成手段12が抽出中の文書構造に組み入れ(ステップS15)、文書構造表示手段13がこのストリーム(内容部)をディスプレイ14の画面に表示されている文書エディタ中に表示する(ステップS16)。次いで、未表

示情報表示手段15が選択されたストリームが上記のストリームセット内の唯一のストリームかを判断し(ステップS17)、ストリームセット内に他のストリームが無いときには処理を終了する。なお、ストリームに含まれるアンカーが下位の階層のストリームセットを参照している場合には、これらストリームセット及びストリームについて上記と同様な選択表示処理が繰り返行われる。

【0088】一方、選択されたストリームの他にストリームセット内に他のストリームが存在する場合には、未表示情報表示手段15が当該他のストリームは選択したストリームより新しいストリームであるかをそのストリーム属性から判断し(ステップS18)、より新しいストリームである場合には当該ストリームを表す候補マークをディスプレイ14上の文書エディタ中に反転表示する一方(ステップS19)、新しいストリームでない場合には当該ストリームを表す候補マークをディスプレイ14上の文書エディタ中に非反転で表示する(ステップS20)。

【0089】ここで、上記のストリーム表示処理(ステップS16)を更に詳しく説明すると、当該処理は図39に示す手順で実行される。まず、選択されたストリームが空のストリームであるかを判断し(ステップS31)、内容部やアンカーといった論理単位を含まない空のストリームである場合にはそのまま処理を終了する。一方、ストリームに論理単位が含まれている場合には、これら論理単位の内未処理のものを1つ選択し(ステップS32)、当該選択した論理単位がアンカーであるかを判断する(ステップS33)。

【0090】この判断の結果、アンカーである場合には図38に示した下位構造抽出表示処理を行ってアンカーで参照されている下位構造の表示を行う一方(ステップS34)、内容部である場合には当該内容部を表示処理する(ステップS35)。そして、選択されたストリーム中の全ての論理単位について上記の処理が終了したかを判断し(ステップS36)、終了していない場合にはステップS32以降の処理を繰り返行い、終了している場合には当該ストリーム表示処理を終了する。

【0091】また、上記のストリーム選択処理(ステップS14)を更に詳しく説明すると、当該処理は図40に示す手順で実行される。まず、ストリームセット中の全てのストリームを候補リストにし(ステップS41)、条件指定手段8から入力された条件にバージョン情報の指定があるかを判断する(ステップS42)。この判断の結果、バージョン指定がある場合には、候補リスト中のストリームから条件に適合する属性を有したストリームを探し、このストリームを一旦基準ストリームとして設定する(ステップS43)。

【0092】次いで、条件指定手段8から入力された条件に基準日時情報の指定があるかを判断し(ステップS

44)、日時指定がある場合には、指定された日時より新しい日時が属性に設定されているストリームを候補リストから削除する(ステップS45)。すなわち、利用者が処理対象としている日時以降に編集処理されたストリームを除外する。次いで、条件指定手段8から入力された条件に利用者情報の指定があるかを判断し(ステップS46)、利用者の指定がある場合には、指定された利用者情報が属性に設定されていないストリームを候補リストから削除する(ステップS47)。すなわち、利用者が処理対象としている自己の編集に係るストリーム以外を除外する。

【0093】次いで、基準ストリームが選択されているかを判断し(ステップS48)、選択されている場合には、基準ストリームよりも古いストリームを候補リストから削除し(ステップS49)、候補リストが空となっていないことを確認して(ステップS50)、基準ストリームを条件指定手段8から入力された条件に適合するストリームとして選択する(ステップS54)。一方、上記の判断の結果、候補リスト中に基準ストリームが残存していない場合には、候補リストが空となってしまっているかを判断し(ステップS52)、空となっている場合には、いずれのストリームの選択されないことを示すために、空のストリームを選択されたストリームとする(ステップS53)。

【0094】また、上記の判断において、基準ストリームが選択されておらず、候補リストは基準ストリームが空ではない場合(ステップS52)又は候補リストから基準ストリームより古いストリームを削除しても空ではない場合(ステップS50)には、候補リスト中の最新のストリームを選択されたストリームとする(ステップS54)。したがって、ストリーム属性に基づいて、条件指定手段8から入力された条件に適合するストリームが選択される。

【0095】次に、利用者が文書エディタ中に表示された候補マーカMa、Mb、・・・を選択することによって、ストリームを置き換えた文書をメモリ1から抽出してディスプレイ14に表示する処理を、図41に示すフローチャートを参照して更に詳しく説明する。まず、論理単位グループ指示手段16によって利用者が候補マーカを指示すると、未表示情報表示手段15が指示されたストリームが属するストリームセットをメモリ1に格納されている文書構造から選択し(ステップS61)、選択したストリームセット中の各ストリームを候補メニューKとして文書エディタ中に表示する(ステップS62)。

【0096】そして、利用者が論理単位グループ指示手段16によって指示して、候補メニューK中の現在選択表示されているストリームとは別のストリームを選択したかを判断し(ステップS63)、別のストリームを選択した場合には、現在文書エディタで表示しているストリ

ームの下位階層の構造を論理単位グループ再選択手段17が破棄し(ステップS64)、更に、再選択されたストリームを破棄したストリームと入れ替える(ステップS65)。そして、この入れ替えられたストリームから成る文書を文書構成手段12が構成し、当該文書を表示手段13が文書エディタ中の表示し直して処理を終了する(ステップS66)。

【0097】次に、編集処理した文書をメモリ1に保存する処理を図42に示すフローチャートを参照して更に詳しく説明する。まず、文書編集手段5によって文書エディタで編集した文書構造のルートアンカーがメモリ1に格納されている文書構造中のルートアンカーリストに含まれているかを判断する(ステップS71)。この結果、ルートアンカーリストにない場合には、編集された文書は全く新規な文書であるので、当該文書構造のルートアンカーを参照関連付手段7がルートアンカーリストに追加する(ステップS72)。

【0098】次いで、文書エディタで生成された内容部を参照関連付手段7が所定のストリームに関連付け(ステップS73)、生成されたストリームを集めて(ステップS74)、以下の処理が未だなされていないストリームを選択する(ステップS75)。次いで、未処理のストリームが対応するストリームセットをメモリ1に格納されている文書構造中から探し(ステップS76)、対応するストリームセットがある場合には(ステップS77)、当該ストリームセットに未処理のストリームを追加する(ステップS79)。一方、対応するストリームセットがない場合には、新たなストリームセットを文書構造中の追加して(ステップS78)、当該ストリームセットに未処理のストリームを追加する(ステップS79)。

【0099】そして、ストリームセットに追加したストリームに対して、日時や利用者といった属性を選択情報対応付け手段4が設定する(ステップS80)。上記の一連の処理は、生成された全てのストリームについて繰り返し行われ(ステップS81)、この後に、ストリーム属性としてのバージョン情報の設定処理が行われる。すなわち、編集保存処理においてバージョン情報の設定がなされているかを判断し(ステップS82)、設定がなされている場合には、メモリ1に格納されている文書構造中の全てのストリームを集めて(ステップS83)、以下の処理を行っていない未処理のストリームを順次選択する(ステップS84)。

【0100】そして、選択したストリームが属するストリームセットを文書構造中から探し(ステップS85)、当該ストリームセット中の選択したストリームの属性にバージョン情報を追加する(ステップS86)。そして、上記ステップS84以降の処理を全てのストリームについて終了したところで、文書保存処理を終了する(ステップS87)。したがって、編集処理で生成さ



れたストリームを保存するに際して、当該ストリームは編集前のストリームと同一のストリームセットに追加され、また、保存処理に際して設定された属性がストリームに対応付けられる。

【0101】なお、上記の実施例で採用したアルゴリズムは、ルートアンカーから順にアンカーに出会ったときに、参照しているストリームセットから選択条件に適合するストリームの選択を逐次行って文書を生成するものであるが、これ以外にも、文書構造に含まれるストリームセットの全てから選択条件に適合する全てのストリームを選択した後で、これら選択したストリームを組み立てて文書を生成するようにしてもよい。また、上記の実施例では、文書エディタの起動時に抽出条件を必ず指定しておくようにしているが、特に指示のないときには、前回の指定と同じにする、或いは、デフォルトの条件にするようにしてもよい。この場合、複数の担当者で自分の編集を一貫して行えることを最も重視するならば、デフォルトの条件は利用者指定を「%本人」にしておけばよい。また、最新の文書の編集を行えることを最も重視するならば、デフォルトの条件は何も指定しないようにすればよい。

【0102】また、上記の実施例では編集処理に際して文書構造中に新しいバージョンを追加するようにしたが、参照処理を行っただけでも、新しいバージョンを追加するようにしてもよい。このようにすれば、参照処理を行った時点での文書構造を取り出すことが可能になる。

【0103】また、上記の本実施例におけるストリームの選択手順ではバージョンを指定した日時よりも基準日時が新しいことを前提としているが、基準日時とバージョン日時が逆転した場合には以下のような方法を採用することにより支障なくストリームを選択処理することができる。

(1) 基準日時よりもバージョンの指定を優先する。

(2) バージョンを指定した日時よりも前の基準日時を指定できないようにする。なお、ストリーム属性ではバージョンを指定した日時は記録されないため、このためには、バージョンを指定した日時を記録する手段が必要である。

【0104】また、上記の実施例ではバージョンを指定した場合にストリーム属性にバージョン情報を追加するようにしたが、この代わりに、新しいストリームとして追加するようにしてもよい。また、バージョンの指定は、利用者が明示的に行っているが、例えば、プリント時等に自動的にバージョンを記録してもよい。また、上記の実施例では、「請求項」等のアンカー名はアンカーを含むストリームに対応付けて管理されている。したがって、アンカーの名前が変更されることによって、新しいストリームが生成されるが、この他にもストリームとは独立にアンカー名を管理したり、参照しているスト

リームの名前として管理するようにしてもかまわない。

【0105】また、上記の実施例では論理単位グループとしてストリームという単位で文書の構造を表現するモデルを採用したが、本発明はこの文書構造の表現方法に限定されるものではなく、従来技術で示したようなノードとリンクからなる木構造による文書構造に本発明を適用することもできる。なお、ノードとリンクによる文書構造においては、本発明の論理単位グループに対応するのは1つのノードを起点とするリンク集合である。要は、本発明は、このリンク集合やストリームといった単位をバージョン管理の対象として、編集処理毎に複製、保存、属性設定等を適切に行う手段を与えていることに特徴がある。

【0106】例えば、従来技術である特開平6-35914号公報に記載された発明を拡張して本発明の機能を提供するためには、次のようにすればよい。末端ではないノードについて下位ノードへのリンク列をノードの内容として扱うことにし、そのノードに対応するバージョンノードのそれぞれが、異なるリンク列を持つようにする。このように拡張すると、下位ノードを本発明における参照単位、下位ノードへのリンク列を論理単位グループとして扱えるようになる。

【0107】更に言えば、本発明では、従来例として説明した多重文書処理システム(特開平7-44563号公報、特開平6-131343号公報)に示されているデータ構造を文書構造として採用することも可能であり、当該データ構造において本発明の論理単位グループに対応するものは、ノードが保持する子ノードポイントの集合となる。なお、当該多重文書処理システムではノードを多重化する手段を提供しているが、これに加えて、本発明は編集時に子ノードポイント集合を多重化する手段を有している。すなわち、本発明では、編集を行った際にノードを多重化するだけでなく、その子ノードポイントの集合を複製、保存、属性設定を行う手段を有している。具体的には、文書の更新が生じたときに、自動的に子ノードポイント集合を複製して管理するとともに、その属性値として日時、利用者、バージョン名を指定できるようにしている。このような処理を利用者が手作業で行うのであれば、編集作業が極めて複雑なものとなることはいうまでもなく、本発明の本質は、この複雑な処理を計算機によって自動的に且つ適切に実行することによって、複数の担当者による執筆作業における負担を大幅に軽減することにある。

【0108】また、上記の実施例では、文書構造を内容部のテキスト表示と一緒に表示する例を示したが、図43、44に示すように、テキスト内容を表示せずに構造だけを表示するようにしてもよい。すなわち、図43では、アンカーを矩形、ストリームを矢印として表示しており、他のストリームの候補があるかどうかは、ストリームを示す矢印を候補の数だけ表示することで示され

る。また、図44では、ストリームを参照するアンカーの木構造として表示しており、ストリームに他の候補があるかどうかは、アンカーを示す図形の右肩に候補の数だけ影をつけることで示される。

【0109】次に、本発明に係る構造化データ処理装置の一実施例を説明する。なお、上記した文書処理装置と比較すると、内容単位はノード、参照単位はリンク、論理単位グループは同一の編集処理に係る（すなわち、兄弟の関係にある）リンクの集合に該当する。また、本実施例についての以下の説明では、過去に存在していなかった構造を生成する例も示す。

【0110】ここで、まず、本実施例が処理の対象としている構造化データのバリエーションについて説明しておく。図45～図47には、ノード、リンク、リンク間関係、文書内容、属性マッピング、選択属性についてのデータ保持方法のバリエーションを示してあり、ノードとリンクからなる従来技術のデータ構造と本質的に異なる点は、同じノードを始点とする複数のリンク間関係を保持し、それぞれに属性情報をマッピングすることによって、リンク間関係を管理することにある。いずれのバリエーションにおいても、これらの構造を表現するために必要な情報が管理されていることが本発明の構成上重要な要件である。

【0111】図45は、リンク間関係の実現方法のバリエーションを示している。同図(a)に示す構造はノードの記述中にそれを始点とする複数のリンク間関係を保持する構成であり、この構造では、リンク間関係は共通の始点ノードを規定する必要がないので、終点ノードへの参照ポインタのリストとして実現できる。なお、本実施例では後述するように、編集対象としてカレント構造管理手段21に保持されるカレント構造を表現するためにこの構成を採用している。また、同図(b)に示す構造はリンク間関係が始点ノードと終点ノードへの参照ポインタのリストとして表現されており、同図(c)に示す構造は共通の始点ノードを持つリンク間関係の集合をリンク間関係セットとして保持し、そこから始点ノードを参照するようになっている。なお、本実施例では後述するように、ハイパー構造管理手段20に保持されるハイパー構造を表現するために、同図(c)に示す構成を採用している。

【0112】図46は、属性マッピングの実現方法のバリエーションを示している。同図(a)に示す構造では、リンク間関係の記述が後述する選択属性を含んでいる。また、同図(b)に示す構造ではリンク間関係セットが各リンク間関係に選択属性をマッピングしており、同図(c)に示す構造では属性値の管理手段が選択属性をリンク間関係にマッピングしている。なお、本実施例では、同図(a)に示す構造を採用している。図47は、ノードによる文書内容の保持方法のバリエーションを示している。同図(a)に示す構造はノード中に内容

を記述する構成であり、同図(b)に示す構造はリンク間関係の記述が内容を記述する構成であり、同図(c)に示す構造は文書内容の管理手段で管理されている内容を参照する構成である。なお、本実施例では同図(b)の構成を採用している。

【0113】次いで、図48には本実施例に係る構造化データ処理装置の構成を示してあり、本装置によって上記構成の構造化データを処理する。構造化データ処理装置には、ハイパー構造管理手段20、カレント構造管理手段21、編集手段22、登録指示手段23、登録手段24、履歴管理手段25、履歴表示手段26、抽出手段27、抽出指示手段28、ルート選択手段29が備えられている。

【0114】ハイパー構造管理手段20はグラフ構造を成している構造化データをメモリに保持して管理しており、この構造化データはノード、リンク、リンク間関係、選択属性、属性マッピングによって構成されている。ノードは構造化データを構成する最小単位であり、文書データを構造化した本実施例では、ノードは文書内容であるテキスト情報を保持している。なお、ノードをリンクで結合した木構造が、文書を表すカレント構造である。リンクは二つのノード間の結合関係を表す単位であり、リンクは始点となるノードと終点となるノードの組として定義される。なお、本実施例では、リンクはリンク間関係を記述する“LinkList”の記述に含まれる。また、このハイパー構造管理手段20は、請求項に言うノード保持手段、リンク保持手段及びリンク関係保持手段を構成している。

【0115】リンク間関係は同じノードを始点とするリンクの集合に対する関係を記述しており、この集合に含まれるリンクのリストと各リンクの位置を規定する。なお、本実施例では、リンク集合は順序を規定されている。選択属性は上記実施例の選択情報に対応した情報であり、同じノードを始点とする一つ以上のリンク間関係の中からリンク間関係を選択するための属性情報である。なお、本実施例では、作成日時、作成者名、バージョン名のリストを、選択属性として用いている。属性マッピングは選択属性をリンク間関係に関連付ける記述であり、本実施例では、リンク間関係を記述する“LinkList”が属性情報を保持する。

【0116】カレント構造管理手段21はノード、リンク、リンク間関係から成る木構造（グラフ構造の一種）で表現されるドキュメント構造データをメモリに保持して管理しており、このドキュメント構造が編集処理の対象となる。なお、本実施例では、リンク間関係およびリンクの記述はノードを記述する“Node”の記述中に含まれる。編集手段22は、ユーザの指示に従ってドキュメント構造に含まれるノード、リンク、リンク間関係についての生成、更新、削除等といった編集処理を行い、候補表示手段30、候補指示手段31、交換手段32を有

している。候補表示手段30は同じノードを始点とするリンク間関係をディスプレイ画面に表示する手段であり、本実施例では、ユーザが選択したノードを始点とするリンク間関係を表示する。候補指示手段31は、ユーザの指示に従って候補表示手段30によって表示されたリンク間関係の中からいずれか一つを選択する手段である。交換手段32は、選択されているリンク間関係をユーザに指示されたリンク間関係に変更する手段である。

【0117】登録指示手段23はカレント構造管理手段21に保持されているドキュメント構造をハイパー構造管理手段20に登録させるための指示を、ユーザから受け取る手段であり、登録手段24は当該指示に従ってドキュメント構造をハイパー構造に登録する。なお、登録はカレント構造中のノード、リンク間関係をハイパー構造中に格納し、適切な属性情報マッピングを行うことである。履歴管理手段25はカレント構造の登録履歴を管理する手段であり、カレント構造管理手段21に保持されているドキュメント構造をハイパー構造管理手段20に登録させる毎に登録履歴を記録して管理する。なお、本実施例では、登録日時、登録者名、登録時に設定したバージョン名の組をセッションとして記録する。履歴表示手段26は履歴管理手段25で管理されているセッションの情報をディスプレイ画面に一覧として表示する手段である。

【0118】抽出手段27は、ユーザからの指示に従って選択されたリンク間関係を用いてハイパー構造管理手段20からドキュメント構造を抽出し、当該ドキュメント構造をカレント構造管理手段21へ保持させる手段であり、検索手段33と選択手段34とを有している。検索手段33は、同じノードを始点とするリンク間関係の中から、ユーザから指示された検索条件を満たすリンク間関係を検索する手段であり、選択手段34は、検索したリンク間関係の中から、ユーザから指示された選択規則に従って1つのリンク間関係を選択する手段である。

【0119】抽出指示手段28はユーザから抽出条件の指示を受け取る手段であり、検索条件指示手段35と選択規則指示手段36とを有している。なお、本実施例では、抽出条件としては選択条件と選択規則の二つが指示可能である。検索条件指示手段35はユーザから選択条件（すなわち、検索条件）の指示を受け取る手段であり、本実施例では、選択条件は日時、バージョン名、執筆者名を含んでいる。選択規則指示手段36はユーザから選択規則の指示を受け取る手段であり、本実施例では、選択規則はユーザ優先、日時優先、バージョン名優先のいずれかである。

【0120】ルート選択手段29は、カレント構造のルートとなるノードをハイパー構造管理手段20に保持されているハイパー構造中から選択する手段であり、本実施例では、ルートノードを起点としてリンク間関係を順次選択することで、カレント構造管理手段21に保持さ

れるカレント構造を生成している。なお、上記した各機能手段20～36は、本実施例ではプロセッサが所定の制御プログラムを実行することにより構成されている。

【0121】ここで、上記した抽出指示手段28、履歴表示手段26、編集手段22は、ユーザに対して情報を提示し或いはユーザが指示や情報等を入力するためのインタフェース画面を表示出力する機能を有している。抽出指示手段28は図49に示すようなディスプレイ画面を表示出力し、この画面には次のような各機能部が設定されている。ルートノード名入力部40はハイパー構造管理手段20から抽出する文書データのルートとなるノード名を入力するためのエリアであり、検索条件入力部41は選択条件を入力するためのエリアである。この検索条件入力部41には上記した選択条件を入力するために、現在使用しているユーザの名前を入力するためのエリアであるユーザ名部42、抽出したい文書データの状態があった日時を入力するためのエリアである日時部43及び44、抽出したい文書のバージョン名を入力するためのエリアであるバージョン名部45が設けられている。

【0122】選択規則入力部46はユーザが選択規則のいずれかを指定するエリアであり、ユーザ優先（USER）を選択すると選択手段34が選択条件を満たすものから指定されたユーザに一致するものを優先的に選択する処理を行い、日時優先（DATE）を選択すると選択手段34が選択条件を満たすものから指定日時を基準として生成日時が最も新しいものを優先的に選択する処理を行い、バージョン名優先（VERSION）を選択すると選択手段34が選択条件を満たすものから指定されたバージョン名を含むものを優先的に選択する処理を行う。また、執筆者選択部47はユーザが選択条件の一部としてユーザ以外の執筆者を選択することができるエリアである。

【0123】また、操作ボタン部48には“extract”ボタンと“history”ボタンとが設けられている。“extract”ボタンを選択すると、指示されたルートノードを始点として選択条件と選択規則に基づいて文書データをハイパー構造管理手段20から抽出する処理が開始され、これと同時に履歴管理手段25が起動される。また、“history”ボタンを選択すると、履歴表示手段26が起動されて、それまでに行われた文書の登録履歴が画面に表示される。図50には、履歴表示手段26が画面表示する登録履歴の例を示してあり、一覧表示された各履歴（すなわち、各行）に一回の登録処理に関する情報として、日時、ユーザ名、バージョン名が表示される。

【0124】編集手段22は図51に示すようなディスプレイ画面を表示出力し、この画面には上記の“extract”ボタンの選択によって生成されて表示出力される。この画面には、文書構造編集ウィンドウ50、文



書内容編集ウィンドウ51、候補選択ウィンドウ52、編集ボタン53、抽出条件表示部54が設けられている。文書構造編集ウィンドウ50は抽出手段27によって抽出された文書データの構造を表示し、ユーザはポインティングデバイスによって文書構造に含まれるノードを選択することができる。

【0125】文書内容編集ウィンドウ51は、文書構造編集ウィンドウ50で選択されたノードの内容を表示し、ユーザは表示しているテキスト内容を編集することによって、選択されたノードの内容を更新することができる。候補選択ウィンドウ52は選択したノードを始点とするリンクリストの候補を表示し、現在選択されているリンクリストを反転して示す。このウィンドウ52において、ユーザはポインティングデバイスによって、他のリンクリストを選択して現在選択しているものと入れ換えることができる。

【0126】編集ボタン53には次の各ボタンが設けられており、ユーザはこれらボタンを用いて文書の編集と保存を行わせることができる。“add child”ボタンを選択すると現在選択されているノードの子ノードが生成され、“add brother”ボタンを選択すると現在選択しているノードの兄弟ノードが生成され、“move down”ボタンを選択すると現在選択しているノードが同じレベルでより後に移動され、“move up”ボタンを選択すると現在選択しているノードが同じレベルでより前に移動され、“move left”ボタンを選択すると現在選択しているノードのレベルが一段上げられ、“move right”ボタンを選択すると現在選択しているノードのレベルが一段下げられ、“move node”ボタンを選択すると現在選択しているノードが削除され、“save”ボタンを選択するとカレント構造管理手段21が保持しているドキュメントがハイパー構造管理手段20が保持しているハイパー構造中に保存され、“close”ボタンを選択すると全体のウィンドウを閉じられて編集処理が完了する。また、抽出条件表示部54にはカレント構造を抽出するための条件として指定された選択条件および選択規則が表示される。

【0127】次に、本実施例に係る構造化データの処理を、図52に示す例を用いてより具体的に説明する。まず、図52には、3名の執筆者、Peter、David、Wickyによる文書の編集活動の履歴を示してあり、図中の横軸は時間の経過を、縦軸は担当者を示している。また、図中の(a)～(e)は、その縦軸の位置に対応する執筆者によって、横軸の位置に対応する日時において、ハイパー構造管理手段20へ保存(登録)された文書データの構造を示している。なお、図53は、編集の履歴を従来技術である導出木によって示したものであり、図中の矢印は導出関係を表しており、導出関係とは始点に位置する文書構造から終点に位置する文書構造が作られた

ことを意味する。このような導出関係は、導出木とよばれる各時点での文書の状態をノードとする木構造を構成する。

【0128】図52では文書構造の木構造はインデントされたテキストによって示しており、図54では(a)～(c)の表している木構造を表現したデータ構造を示している。図55～図57は、(a)～(c)の文書構造の保存が行われたときのハイパー構造の状態を示している。この例は、“WriteAtWill”という製品のマニュアルをPeter、David、Wickyの3人で執筆している例である。以下順を追って作業の経緯を説明する。

【0129】まず、(Aug 1)において、Peterが“WriteAtWill”のマニュアルを書き始め、この最初の状態である文書データ(a)を“PC version”という名前で保存した。次いで、(Aug 2)に、DOSとWINとの2つのOS用の記述が必要になったので、文書データ(a)の章立てをDOSとWINとに分けた文書データ(b)を作成し、名前を付けずに保存した。その後、2つの異なるマニュアルをそれぞれ別の担当で執筆することが決まり、それぞれDavidとWickyとに担当が割り当てられた。そして、(Aug 3)に、Davidが文書データ(a)の中からDOS用の記述を取り出して加筆した文書データ(c)を作成し、“DOS version”という名前で保存した。更に、(Aug 4)に、Wickyが文書データ(b)の中からWIN用の記述を取り出して加筆した文書データ(e)を作成し、“WIN version”という名前で保存した。そして、(Aug 5)に、Davidが文書データ(c)にさらに加筆した文書データ(d)を作成し、名前を付けずに保存した。

【0130】このような場合において、本実施例では過去に存在していなかった文書データを生成することができる。すなわち、本発明は、従来技術のように過去に存在していた文書のバージョンが復元できるだけでなく、存在していなかった文書の構造を取り出せる。図58の文書データ(g)及び(f)は、図53に示した導出木上にはない文書データである。この文書データ(g)の構造は、Wickyの(Aug 4)の構造の内の“INTRO”以下の部分をDavidが(Aug 3)に加筆したものに置き換えて得られる構造である。すなわち、文書データ(g)は、Wickyにとっては自分の書いた部分はそのままにして、他の執筆者の書いた最新の情報を反映させたものである。また、文書データ(f)は、Peterが(Aug 2)に定義した全体の枠組みの中に、他の執筆者の書いた最新の情報を組み込んだものである。したがって、本発明では、このような過去に存在せず、また、予定もしていなかった新しい文書構造を、過去に作られた文書の部分を利用して簡単に取り出すことが可能となっている。

【0131】ここで、(Aug 6)の時点で文書を編集するときの選択条件と抽出結果のバリエーションを示

す図59～図74を参照して、文書抽出の処理をより具体的に説明する。図59には履歴表示手段26で表示される(Aug 6)以前に行われた文書保存の履歴を示してあり、図60にはDavidによって指定された選択条件を示してある。図60に示す画面上において、ルートノード名入力部40にはルートノードの名前としてマニュアルのタイトルである"WAW Manual"が入力指定され、ユーザ名入力部42にはユーザの名前として"David"が入力指定され、日時入力部43にはいつまでに作成された内容を探索するかを示す日時の情報として日付"Aug/06/96"及び時間"19:03"が入力指定され、バージョン名入力部45には検索を開始する起点となるバージョンを示すバージョン名として"DOS version"が入力指定されている。

【0132】なお、ユーザ以外のどの執筆者が作成した内容を探索するかを指定する執筆者選択部47には、この例では、誰も指定していないのでユーザであるDavidが作成した文書内容が検索される。また、この例では、選択規則入力部46で"USER"が選択されており、自分で作成した内容を優先して選択することが指示されている。

【0133】図61には、図60に示した条件によってハイパー構造管理部20から文書データを抽出した結果を示してある。なお、この例では、図52に示した文書データ(d)が抽出されている。また、図60に示した条件を変更して、図62に示すように日時入力部43の条件を"Aug/03/96"に指定すると、(Aug 3)までで最新の内容を反映した文書データの抽出が指示される。この結果、図63に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図52に示した文書データ(c)である。

【0134】また、図60に示した条件を変更して、図64に示すようにユーザ入力部42の指定を"Wicky"とし、バージョン名入力部45の指定をバージョン名"WIN version"とすると、図65に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図52に示した文書データ(e)である。また、図64に示した条件を変更して、図66に示すように選択規則入力部46の指定を"DATE"にし、さらに、執筆者選択部47の指定を他の執筆者としてDavidとWickyを加えるように指定すると、図67に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図58に示した文書データ(g)であり、図52に示した導出木の中には存在しない、従来技術では選択できなかった

データ構造が抽出されている。

【0135】また、図68に示すようにユーザ名入力部42の指定を"Peter"にし、さらにバージョン名入力部41の指定を"PC version"に変更すると、図69に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図52に示した文書データ(b)である。また、図68に示した条件を変更して、図70に示すように選択規則入力部46の指定を"VERSION"とすると、図71に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図52に示した文書データ(a)である。

【0136】また、図68に示した条件を変更して、図72に示すように、執筆者選択部47の指定を他の執筆者としてDavidとWickyを加えるように指定すると、図73に示すような文書構造がハイパー構造管理部20から抽出されて画面表示される。なお、この抽出された文書データは、この例では、図58に示した文書データ

(g)であり、図52に示した導出木の中には存在しない、従来技術では選択できなかったデータ構造が抽出されている。ここで、図73に示す画面では、さらに最上位のレベルで入れ換え可能なリンクリストの候補が候補選択ウィンドウ52に表示され、また、現状の選択状況が反転表示されている。この例では、4つの選択肢の中からWickyの作成した、"Win version"のバージョン名のリンクリストが選択されていることを反転表示で示している。

【0137】また、図74には、図73において選択肢として示されているリンクリストから過去にPeterが作成したものを選択し、現状のものと入れ換える指示を行った状態を示してある。この結果、図58における文書データ(f)の構造が抽出され、図52に示した導出木の中には存在しない、従来技術では選択できなかったデータ構造が抽出される。

【0138】次に、本実施例のデータ構造とアルゴリズムについて説明する。本実施例は"Windows95"（登録商標）上で、"Java"（登録商標）言語によって実現されている。"Java"はオブジェクト指向言語であり、オブジェクトと呼ばれるデータ構造と、そのオブジェクトが受け付け実行するメソッドと呼ばれる手続きを、クラスの定義として記述する。以下に示すプログラムリストは、本実施例を実現しているJavaプログラムから主要な部分を抜粋、簡略化したものであり、このプログラムリストを参照しながらデータ構造とアルゴリズムを説明する。なお、データ構造は図75に示す。

【0139】

(1) リンク間関係記述の集合"LinkListHolder"のクラス定義

```
「1」 public class LinkListHolder extends Object{
```

```
// LinkListHolderは、同じノードを始点に持つリンクリストの集合を管理す
```

るオブジェクト、

/\* データ構造の定義 \*/

「2」 Node sourceNode;

//始点となるノード

「3」 Vector linkListSet;

//リンクリストの集合

/\* メソッドregisterLinksの定義 \*/

「4」 void registerLinks(Node node,String user, Date date, String version) {

//指定されたノードを始点とするリンクリストをリンクリストセットに登録(追加)するメソッド

【0140】

「5」 int index = -1;

//リンクリストセット中の同じリンクリストの番号を格納する変数

「6」 LinkList linkList;

//リンクリストセット中の同じリンクリストを格納する変数

「7」 for (int i=0; i<linkListSet.size();i++) {

//リンクリストセットに登録されている中から同じリンクリストを探す

「8」 linkList = (LinkList)(linkListSet.elementAt(i));

「9」 if (linkList.isLinksFor(node)) {

「10」 index = i;

「11」 break;

「12」 }

「13」 }

「14」 if (index >= 0) {

//同じリンクリストが見つかったとき

【0141】

「15」 if (version.length() == 0) {

//バージョン名が指定されたかをチェック

「16」 } else {

「17」 linkList.addVersion(version);

//見つかったリンクセットにバージョン名を追加

「18」 }

「19」 }else {

//同じリンクリストが見つからないとき

「20」 linkList = new LinkList(node,user,date,version);

//nodeのもつリンクリストにユーザ名、日時、バージョン名を対応付けたリンクリストを生成

【0142】

「21」 addLinkList(linkList);

//新たに生成したリンクリストを追加する、

「22」 }

「23」 }

/\* メソッドselectLinksの定義 \*/

「24」 Vector selectLinks(String user, Vector authors,Date date,String version,int rule) {

//抽出条件を満たすリンクのリストから指定された規則(rule)にしたがって一つを選択するメソッド、LinkListSetは登録日時の新しい順に並ぶように登録しているので、最初に見つかったものが最も新しいリンクリストになる、

「25」 int userindex = -1;

//ユーザ名が合致する最初のリンクリストの番号を格納する変数

【0143】

```
「26」      int dateindex = -1;
            //日時の条件が合致する最初のリンクリストの番号を格納する変数
「27」      int versionindex = -1;
            //バージョン名が合致する最初のリンクリストの番号を格納する変数
「28」      for (int i=0; i<linkListSet.size(); i++) {
            //リンクリストセット中のすべてのリンクリストを探索
「29」          LinkList linkList = (LinkList)linkListSet.elementAt(i);
「30」          if (linkList.satisfies(authors,date,version)) {
            //執筆者、日時、バージョン名に関する条件を満たすものについて以下を実行
```

【0144】

```
「31」          if (dateindex < 0) {
            // まだ日時の条件を満たすものがみつかっていなければ
「32」              dateindex = i;
            // このリンクリストを最新のものとする
「33」          }
「34」          if (linkList.regUser.equals(user)) {
            // ユーザ名に合致するものがあれば
「35」              if (userindex < 0) {
            // まだユーザ名に合致するものがみつかっていなければ
```

【0145】

```
「36」              userindex = i;
            // このリンクリストを最新のものとする
「37」          }
「38」          }
「39」          if (linkList.hasVersion(version)) {
            // バージョン名が合致するものがあれば
「40」              if (versionindex < 0) {
            // バージョン名が合致するものがみつかっていなければ
```

【0146】

```
「41」              versionindex = i;
            // このリンクリストを最新のものとする
「42」          }
「43」          }
「44」          }
「45」          }
```

【0147】

```
「46」      int index = -1;
            // 選択するリンクリストの番号を格納する変数
「47」      if (rule == DATE) {
            // 選択規則が日時最新のとき
「48」          index = dateindex;
            // 日時が最新のものを選択
「49」      } else if (rule == USER) {
            // 選択規則がユーザ名優先のとき
「50」          if (userindex < 0) {
            // ユーザ名に合致するものがなければ
```

【0148】

```
「51」          index = dateindex;
```

```

// 最新のものを選択
「52」         } else {
// ユーザ名に合致するものがあれば
「53」             index = userindex;
// ユーザ名に合致する最新のものを選択する
「54」         }
「55」     } else if (rule == VERSION) {
// 選択規則がバージョン名優先のとき

```

【0149】

```

「56」         index = versionindex;
// バージョン名が合致する最新のものを選択する
「57」     }
「58」     if (index < 0) {
// もし選択されたものがなければ、
「59」         return new Vector();
// 空のリストを返す
「60」     } else {

```

【0150】

```

「61」         return ((LinkedList)linkListSet.elementAt(index)).links;
// 選択されたリンクリストを返す
「62」     }
「63」 }
「64」 }

```

【0151】

```

(2) ノード記述Nodeのクラス定義
「65」 public class Node extends Object {
// ノードを記述するオブジェクト
/* データ構造 */
「66」     String contents;
// ノードの保持するテキスト内容
「67」     Vector links;
// ノードを始点とするリンクのリスト
「68」     Long nodeID;
// ノードの識別子. この値が一致するノードを同一のノードと見なす.
「69」 }

```

【0152】

```

(3) リンク間関係記述LinkedListのクラス定義
「70」 public class LinkedList extends Object {
// 同じノードを始点とするリンクのリストを表すオブジェクト
/* データ構造の定義 */
「71」     String sourceNodeContents;
// ソースノードの内容
「72」     Vector links;
// 終点ノードのリスト
「73」     Date regDate;
// 作成日時
「74」     String regUser;
// 作成者
「75」     Vector versions;
// バージョン名のリスト

```

【0153】

```

/* メソッドisLinksForの定義 */
「76」    public boolean isLinksFor(Node node) {
        // 指定されたノードのリンクリストに一致するかを調べるメソッド.
「77」        if (!sourceNodeContents.equals(node.contents)) return false;
        // ソースノードと内容が違えば一致していない
「78」        if (links.size() != node.links.size()) return false;
        // リンクリストの長さが違えば一致しない
「79」        boolean equal = true;
        // 一致しないものが一つでもあるかどうかを調べる
「80」        for (int i=0; i < links.size(); i++) {

```

【0154】

```

「81」            equal = ((Node)links.elementAt(i)).equals((Node)node.links.elementAt(i));
「82」            if (!equal) break;
「83」        }
「84」        return equal;
「85」    }

```

【0155】

```

/* メソッドsatisfiesの定義 */
「86」    blic boolean satisfies(Vector authors, Date date, String version)
    {
        // 条件を満たすかどうかを調べるメソッド
「87」        if (regDate.after(date)) {
            // 登録日時が指定された日時よりも後であれば満たさない
「88」            return false;
「89」        }

```

【0156】

```

「90」        for (int i=0; i<versions.size(); i++) {
            // 指定されたバージョン名を含んでいれば満たす
「91」            if (((String)versions.elementAt(i)).equals(version)) {
「92」                return true;
「93」            }
「94」        }
「95」        for (int i = 0; i < authors.size(); i++) {
            // 指定された執筆者を含んでいれば満たす.

```

【0157】

```

「96」            String au = (String)authors.elementAt(i);
「97」            if (regUser.equals(au)) {
「98」                return true;
「99」            }
「100」        }
「101」        return false;
「102」    }
「103」 }

```

【0158】

```

(4) ハイパー構造管理手段HyperStructureManagerのクラス定義
「104」 public class HyperStructureManager extends Object {
        // ノードとリンクからなる全体構造を管理するオブジェクト
/* データ構造の定義 */

```

```

「105」    Vector linkListHolderSet;
           // 含まれるリンクリストの集合
「106」    Vector regVersions;
           // 登録されているバージョン名のリスト
「107」    Vector regAuthors;
           // 登録されている著者名のリスト
「108」    Vector sessionHistory;
           // 実行された登録操作の履歴

```

【0159】

```

/* メソッドsaveDocumentの定義 */
「109」    void saveDocument(Node root, String user, Date date, String ve
rsion) {
           // ルートノードによって代表されるカレント構造を登録するメソッド
「110」        if (root == null) {
「111」            } else if (regVersionsHas(version)) {
           // 既にバージョン名が登録されていればなにもしない。
「112」            } else {
「113」                if (version.length() > 0) {
           // バージョン名が指定されていれば
「114」                    regVersions.addElement(version);
           // 指示されたバージョン名を登録する
「115」                }

```

【0160】

```

「116」        saveTree(root,user,date,version);
           // ルートノードからトラバースできる構造をハイパー構造に登録する
「117」        sessionHistory.addElement(new Session(user, date, vers
ion));
           // 登録操作の履歴を追加
「118」        }
「119」    }

```

【0161】

```

/* メソッドsaveTreeの定義 */
「120」    void saveTree(Node node, String user, Date date, String versio
n) {
           // 指示されたノードからトラバースできる構造をハイパー構造に登録する
           // メソッド
「121」        int index = -1;
           // ハイパー構造からノードを始点とするリンクリストホルダーを探す
「122」        for(int i=0; i<linkListHolderSet.size();i++) {
「123」            if (((LinkListHolder)linkListHolderSet.elementAt(i)).s
ourceNode.equals(node)) {
「124」                index = i;
「125」                break;

```

【0162】

```

「126」        }
「127」    }
「128」    LinkListHolder llholder;
           // 指示されたノードを始点とするリンクリストホルダーを格納する変数
「129」    if(index < 0) {
           // リンクリストホルダーが見つからなければ

```

【0163】

```

「130」        llholder = new LinkedListHolder(node);
                // 新しいリンクリストホルダーを生成して選択する

「131」        linkListHolderSet.addElement(llholder);
                // 生成したリンクリストホルダーを追加する
「132」        } else {
「133」        llholder = (LinkedListHolder)linkListHolderSet.elementAt
(index);
                // 見つければそのリンクリストホルダーを選択する。
「134」        }
「135」        llholder.registerLinks(node,user,date,version);
                // リンクリストホルダーに指示されたノードのリンクリストを追加する

```

【0164】

```

「136」        for (int i = 0; i < node.links.size(); i++) {
「137」        saveTree((Node)node.links.elementAt(i),user,date,versi
on);
                // 指示されたノードのリンクの終点のノードについてsaveTreeメソッドを
再帰的に呼び出す
「138」        }
「139」    }

```

【0165】

```

/* メソッドextractDocumentの定義 */
「140」    void extractDocument(Node root, String user, Vector authors, D
ate date, String version, int rule) {
                // ルートノードとして指定されたノードを起点にして文書を表すカレント
構造を抽出するメソッド
「141」        LinkedListHolder linkListHolder = linkListHolderFor(root);
                // ルートノードを始点とするリンクリストホルダーを選択する
「142」        root.contents = linkListHolder.sourceNode.contents;
                // ルートノードの内容としてリンクリストホルダーの内容を読み込む
「143」        Date sesdate = sessionDateOf(version);
                // バージョン名が登録された日時を取り出す
「144」        if (sesdate != null) {
「145」        if (date.before(sesdate)) {
                // 指定された日時がバージョン名の登録日よりも以前かどうかを調べる

```

【0166】

```

「146」        } else {
「147」        extractTree(root, user, authors, date, version, ru
le);
                // 登録日以降の指定日のときにルートノードを起点として構造を抽出する
「148」        }
「149」    }
「150」}

```

【0167】

```

/* メソッドextractTreeの定義 */
「151」    void extractTree(Node node, String user, Vector authors, Date
date, String version, int rule) {
                // 指定されたノードを起点として構造を抽出するメソッド
「152」        LinkedListHolder llholder = linkListHolderFor(node);
                // 指定されたノードを始点とするリンクリストホルダーを選択する

```



```

「153」      Vector links = llholder.selectLinks(user, authors, date, v
ersion, rule);
      // リンクリストホルダーの中から抽出条件を満たすリンクを選択する
「154」      node.links = new Vector();
      // ノードのリンクリストを初期化
「155」      Node nd = null;

【0168】
「156」      for (int i=0; i < links.size(); i++) {
      // 選択されたリンクリストの各リンクについて以下を実行
「157」          nd = ((Node)links.elementAt(i)).copy();
      // リnkの終点ノードをコピーして新しいノードを生成する
「158」          node.addNode(nd);
      // 生成したノードを指定されたノードのリンクリストに追加する
「159」          extractTree(nd, user, authors, date, version, rule);
      // 生成したノードを起点にしてextractTreeを再帰的に実行する
「160」      }
「161」    }

```

【0169】また、図75には、本実施例の主要な構成要素のデータ構造を示してある。同図(a)に示すカレント構造は、本実施例では、編集対象の文書を表示するデータ構造であり、このカレント構造をノード記述"Node"からなる木構造の形で表現している。この"Node"のデータ構造は上記したリストの65行目から68行目までが対応するオブジェクトの定義であり、次の記述を含んでいる。"contents"は文書内容を有しており、本実施例では文字列である。"links"は1つ以上の"Node"への参照のリストを有しており、このリストはこの"Node"を始点とするリンクの集合と順序関係を定めるリンク間関係記述である。"nodeID"はノードの識別子を有しており、この値によってノードが同一であるかどうか判断される。

【0170】図75(b)に示すハイパー構造は、ハイパー構造管理手段20で保持管理される構造化データであり、本実施例では、このハイパー構造を"LinkedList"、"LinkedListHolder"によって構造化された"Node"のネットワーク構造の形で表現している。文書データの構造は、"Node"の"links"による他の"Node"への参照を用いて構成されるが、ハイパー構造中では"LinkedList"と"LinkedListHolder"からの参照によって構成される。このハイパー構造は次の記述を含んでいる。"LinkedList"は、同じノードを始点とするリンクの集合とその間の関係を記述するリンク間関係記述であり、さらに、このリンク間関係記述に対応づける選択属性も管理する。なお、上記したリストの70行目から75行目までが対応するオブジェクトの定義である。

【0171】また、"sourceNodeContents"はリンクの始点となるノードの文書内容を有しており、"links"はリンクの終点となるNodeへの参照のリストを有している。また、"regDate"は選択属性である作成日時を有し、"regUser"は選択属性である作成者名を有し、"v

ersions"は選択属性であるバージョン名のリストを有している。"LinkedListHolder"は、同じNodeを始点とするLinkedListの集合を保持しており、上記したリストの1行目から3行目までが対応するオブジェクトの定義である。また、"sourceNode"は始点となるNodeを参照し、"linkListSet"は同じNodeを始点とするLinkedListの集合を有している。

【0172】ハイパー構造管理手段20が保持管理するデータ構造は、図75(c)に示してあり、本実施例では、ハイパー構造管理手段は"HyperStructureManager"というオブジェクトとして実現されている。"HyperStructureManager"は以下の情報を管理しており、上記したリストの65行目から68行目までが対応するオブジェクトの定義である。"LinkedListHolderSet"は"LinkedListHolder"の集合を有し、"regVersions"はカレント構造の登録時に登録されたすべてのバージョン名のリストを有し、"regAuthors"はこの構造を編集できる執筆者の名前のリストを有し、"sessionHistory"はこの構造に対して行った文書登録操作の履歴情報を有している。

【0173】次に、上記したリストを参照して本実施例のアルゴリズムを説明する。"HyperStructureManager"のメソッドは、上記リストの140行目から150目に対応する"extractDocument"、151行目から161行目に対応する"extractTree"、109行目から119行目に対応する"saveDocument"、120行目から138行目に対応する"saveTree"を有している。"extractDocument"はルートとして指示されたノードを起点にして、カレント構造管理手段21に保持するカレント構造をハイパー構造管理手段20に保持されたハイパー構造から抽出する処理を行う。この抽出処理では、ルートから開始して、下位に位置するノードについて下位構造を順次抽出するメソッド"extractTree"を呼び

出すことによって木構造を抽出する。

【0174】また、“extractTree”は、指定されたノードを起点としてデータ構造を抽出する処理を行い、指定されたノードを始点とするリンクリストホルダーから抽出条件を満たすリンクリストを選択し、その内容を持つノードを生成し、さらにその参照先のノードについて“extractTree”を再帰的に実行する。“saveDocument”は、ルートノードによって代表されるカレント構造をハイパー構造に登録する処理を行い、ルートノードが参照する下位のノードについて、下位構造に登録するメソッド“saveTree”を呼び出すことによって、木構造全体に登録する。“saveTree”は、指定されたノードの下位構造をハイパー構造に登録する処理を行い、指定されたノードを始点とするリンクリストホルダーにノードが保持するリンクリストに一致するものがなければ、リンクリストを追加し、さらにその下位ノードについて“saveTree”を再帰的に実行する。

【0175】次に、“LinkListHolder”のメソッドは、上記リストの4行目から23行目に対応する“registerLinks”、24行目から64行目に対応する“selectLinks”を有している。“registerLinks”は、指定されたカレント構造中のノードが持つリンクのリストを追加する処理を行い、追加対象のリンクリストと一致するものがすでに登録されているかを調べ、登録されていないければ、新しいリンクリストとして追加する。なお、すでに登録されていて、かつ、バージョン名が指定されているときには、バージョン名の設定を行う。“selectLinks”は、保持しているリンクリストの中から選択条件を満たすリンクリストをすべて検索し、その中から選択規則に従ってリンクリストを1つ選択する処理を行う。

【0176】次に、“LinkList”のメソッドは、上記したリストの76行目から85行目に対応する“isLinksFor”と、86行目から103行目に対応する“satisfies”を有している。“isLinksFor”は、カレント構造中のノードの持つリンクのリストと一致するかどうかを判断する処理を行い、ノードの文書内容と順序も含めたリンクの集合が一致するときに“True”を返す。“satisfies”は、選択条件を満たすかどうかを判断する処理を行い、指定された日時を含んでそれ以前に作成されたもので、指定されたバージョン名を含んでいるか、または、指定された執筆者によって作成されたものであれば“True”を返す。

【0177】上記のアルゴリズムにより、抽出指示手段28から指定された選択条件及び選択規則に従って、該当する文書構造が抽出手段27によってハイパー構造管理手段20から抽出され、抽出された文書構造が編集手段22による編集処理の対象としてカレント管理手段21に保持される。また、編集処理がなされた文書構造は登録手段24によってハイパー構造管理手段20へ登録され、この登録処理に際して履歴情報が履歴管理手段2

5によって登録される。

【0178】次に、本発明の構造化データ処理装置を図形の編集に適用した他の一実施例を説明する。なお、本実施例の構成、データ構造、アルゴリズムは、本質的に上記した構造化データ処理装置の実施例と同じであるので、重複する説明は割愛する。図76には、図形の編集過程の一例を示してあり、横軸に日時を、縦軸に執筆者名を示し、図中の(a)(b)(c)で示す図形が、縦軸の位置に対応する執筆者による編集結果として横軸の位置に対応する日時にハイパー構造管理手段20に保存されたことを示している。また、この例では、(Aug 1)にPeterが(a)に示す図形を作成し、保存した。そして、(Aug 2)にDavidが(b)に示すように、(a)に含まれる図形要素である文字列“蓄積部”の部分データベースの概念図を含むように変更し、保存した。さらに、(Aug 3)にPeterが(c)に示すように、(a)に新しい図形要素である文字列“計算部”を加えて保存した。

【0179】図77には、上記の(a)(b)(c)それぞれのデータ構造を示してあり、上記した実施例と異なる点は、文書内容として図形要素を持ち、リンク間関係が図形要素中の参照位置として二次元位置を規定していることである。それぞれの図(a)(b)(c)で、(1)はデータの論理的な構造を図示したものであり、(2)は文書内容であると同時にリンク間の関係を記述するために使われている図形要素との関係を含めて図示したものである。また、図78の(a)には、ハイパー構造管理手段20に保持されるハイパー構造中のデータ構造を示してある。また、同図の(b)には選択条件を示してあり、当該選択条件を指示することによって、図79の(a)(b)(c)に示す図形がカレント構造として抽出されてカレント構造管理手段21に保持される。なお、ここに示す抽出されたデータ構造は、PeterとDavidの両者の編集の最新状態を取り込んだものであり、過去には存在しなかった構造である。

【0180】次に、本発明の構造化データ処理装置をハイパーテキストの検索に適用した更に他の一実施例を説明する。なお、本実施例では、編集ではなく、検索履歴として生成される木構造を処理の対象にする点が異なるが、本質的なデータ構造やアルゴリズムは上記した実施例と同じである。

【0181】図80には、ある調査のために、Peterが(Aug 1)に行ったネットワーク検索の一例を示してある。すなわち、まず、http://AAA.htmlというアドレスのページを開いて、そこから関連するページBBBへのアンカーを選択してジャンプした。この結果、http://BBB.htmlというアドレスのページが開き、さらに、そこから関連するページDDDへのアンカーを選択してジャンプした。この結果、http://DDD.htmlというアドレスのページが開き、さらに、そこから前のページに戻る指

示をした。この結果、http://BBB.htmlというアドレスのページが開き、さらに、そこから関連するページEEEへのアンカーを選択してジャンプし、その結果として、http://EEE.htmlというアドレスのページが開いた。

【0182】図81には、同じ目的の調査のために、Davidが(Aug 2)に行ったネットワーク検索の一例を示してある。すなわち、まず、http://AAA.htmlというアドレスのページを開いて、そこから関連するページBBBへのアンカーを選択してジャンプした。この結果、http://DDD.htmlというアドレスのページが開き、さらに、そこから前のページに戻る指示をした。この結果、http://BBB.htmlというアドレスのページが開き、さらに、そこから関連するページCCCへのアンカーを選択してジャンプした。この結果、http://CCC.htmlというアドレスのページが開き、さらに、そこから関連するページFFFへのアンカーを選択してジャンプした。この結果、http://FFF.htmlというアドレスのページが開き、さらに、そこから前のページに戻る指示をした。この結果、http://CCC.htmlというアドレスのページが開き、さらに、そこから関連するページGGGへのアンカーを選択してジャンプし、この結果として、http://GGG.htmlというアドレスのページが開いた。

【0183】図82の(a)(b)には、上記したそれぞれの検索の履歴を木構造として表現してある。検索を開始したノードをルートにして、検索に利用したリンクだけを選択することによって、検索したデータ空間を表す木構造が生成される。なお、ハイパーテキストの検索履歴からこのような木構造を得るアルゴリズムとしては、特開平6-259312号公報に記載されている従来技術を採用している。なお、同図の(c)には、検索の履歴の構造をカレント構造として、ハイパー構造に登録した結果を示してある。なお、本実施例では、ハイパー構造は、WWWなどの分散したネットワーク構造の複数の部分構造を組み合わせて得られるものである。ハイパー構造は、検索を実行するクライアントのための情報として、ローカルに保持されるようにしてもよいし、ネットワークを維持している無数のサーバに分散して保持されるようにしてもよい。

【0184】図83の(a)には、ハイパー構造管理手段20からカレント構造を抽出するために指定した選択条件を示してあり、同図の(b)には、抽出されたカレント構造を示してある。この抽出されたカレント構造は、PeterとDavidの行った調査をマージした調査全体の検索空間を表わす過去に存在しなかった構造である。

【0185】なお、本発明は、上記した実施例以外に、例えば、部品のバージョンと全体構造のバージョンの両方を管理し、柔軟に組み合わせることに意味のあるような様々なアプリケーションに適用することもできる。また、本発明を適用することによって、それぞれの視点で書かれた新しい情報を制約なく反映させることができ、

また、過去の情報構造の断片を生かして新しい状況に合わせた構造を生成することができる。例えば、分散した環境での電子メールや電子掲示板での意見交換に適用することによって、過去の議論の復元や、過去の議論の断片から新しい情報の構造を構築することができる。

【0186】また、書いた人がお互いを知らないほど多数の文書を管理するシステムに適用することによって、コメントや修正によって直接文面を修正しても支障が起らず、さらに、新しい文書を合成する機能を持つ文書データベースを実現することができる。また、電子メールなどの流れの構造や組織の構造などの管理に本発明を適用することによって、過去の仕事の依頼の流れの復元や、それを用いた業務の分析や設計を行う機能を実現することができる。また、担当者が複数いるときばかりでなく、一人で文書を編集するシステムに適用した場合でも、当初想定していなかった目的のために、それまでに作ってきた文書から適切な部分を抜き出して必要な文書に再構成する機能を実現することができる。

【0187】

【発明の効果】以上説明したように、本発明に係る構造化データ処理装置によると、ノード及びリンクから構成されたグラフ構造の構造化データについて、リンク間関係保持手段に同一のノードを始点とする複数のリンクの内の同一の編集処理に係るリンクを集合として保持させるとともに各集合中に含まれるリンク間の位置関係情報を保持させるようにしたため、文書データ等のグラフ構造をなす構造化データについて、利用者に自由に編集をさせても、過去のデータ構造の復元や新しい状況に合わせた適切なデータ構造の作成を実現することができる。また、本発明は、利用者による簡易な指定操作によって、過去に作ってきたデータを元にして、当初想定していなかった新しい状況に合わせた適切な構造化データを作り出すことを実現することができる。

【0188】特に、本発明に係る文書処理装置及び方法によると、論理単位を含む論理単位グループに階層構造によって文書構造を構成し、編集処理に際しては新たな論理単位グループを生成して元の論理単位グループと同一の集合として保持するようにしたため、文書の全体構造を編集によって変更した場合であっても、編集前の構造で文書を復元させることができる。したがって、執筆者に編集上の制約を与えずに文書を自由に編集させても過去の文書を支障無く復元することができ、共同執筆を支障無く実現させることができる。また、本発明に係る文書処理装置及び方法によると、各論理単位グループに選択情報を対応付けるようにしたため、利用者が選択情報を含む選択条件を指定するといった簡易な操作によって、適合する過去の文書バージョンを復元させることができ、任意のバージョンの文書について参照や編集処理を容易に行うことができる。

【図面の簡単な説明】

【図1】 本発明の一実施例に係る文書処理装置の構成図である。

【図2】 ストリームセットの構成を示す概念図である。

【図3】 文書構造の一例を示す概念図である。

【図4】 内容部が変更された文書構造の一例を示す概念図である。

【図5】 部分構造が挿入された文書構造の一例を示す概念図である。

【図6】 部分構造が削除された文書構造の一例を示す概念図である。

【図7】 部分構造の順所が入れ替えられた文書構造の一例を示す概念図である。

【図8】 部分構造が移動された文書構造の一例を示す概念図である。

【図9】 ストリーム選択の原理を説明する概念図である。

【図10】 文書の編集セッションの一例を示す図である。

【図11】 セッション1において起動された文書エディタを示す図である。

【図12】 セッション1において編集作業中の文書エディタを示す図である。

【図13】 セッション1において生成された文書構造の一例を示す図である。

【図14】 セッション1において保存処理のツールを示す図である。

【図15】 セッション1においてメモリに格納された文書構造の一例を示す図である。

【図16】 セッション2において文書フォルダ及び抽出条件設定ツールを示す図である。

【図17】 セッション2において編集作業中の文書エディタを示す図である。

【図18】 セッション2において生成された文書構造の一例を示す図である。

【図19】 セッション2においてメモリに格納された文書構造の一例を示す図である。

【図20】 セッション3において編集作業中の文書エディタを示す図である。

【図21】 セッション3において生成された文書構造の一例を示す図である。

【図22】 セッション3においてメモリに格納された文書構造の一例を示す図である。

【図23】 セッション4において起動された抽出条件設定ツールを示す図である。

【図24】 セッション4において編集作業中の文書エディタを示す図である。

【図25】 セッション4において生成された文書構造の一例を示す図である。

【図26】 セッション4において保存処理のツールを

示す図である。

【図27】 セッション4においてメモリに格納された文書構造の一例を示す図である。

【図28】 セッション5において編集作業中の文書エディタを示す図である。

【図29】 セッション5において生成された文書構造の一例を示す図である。

【図30】 セッション5においてメモリに格納された文書構造の一例を示す図である。

【図31】 セッション6において起動された抽出条件設定ツールを示す図である。

【図32】 セッション6において参照作業中の文書エディタを示す図である。

【図33】 セッション6において生成された文書構造の一例を示す図である。

【図34】 セッション7において編集作業中の文書エディタを示す図である。

【図35】 セッション7において保存処理のツールを示す図である。

【図36】 セッション7においてメモリに格納された文書構造の一例を示す図である。

【図37】 文書抽出表示処理の手順の一例を示すフローチャートである。

【図38】 下位構造抽出表示処理の手順の一例を示すフローチャートである。

【図39】 ストリーム表示処理の手順の一例を示すフローチャートである。

【図40】 ストリーム選択処理の手順の一例を示すフローチャートである。

【図41】 文書再選択表示処理の手順の一例を示すフローチャートである。

【図42】 文書保存処理の手順の一例を示すフローチャートである。

【図43】 文書構造の表示形態の他の一例を示す図である。

【図44】 文書構造の表示形態の更に他の一例を示す図である。

【図45】 本発明の構造化データ処理に係るデータ構造のバリエーションを示す概念図である。

【図46】 本発明の構造化データ処理に係るデータ構造のバリエーションを示す概念図である。

【図47】 本発明の構造化データ処理に係るデータ構造のバリエーションを示す概念図である。

【図48】 本発明に係る構造化データ処理装置の一実施例を示す構成図である。

【図49】 抽出指示手段の画面表示の一例を示す図である。

【図50】 履歴表示手段の画面表示の一例を示す図である。

【図51】 編集手段の画面表示の一例を示す図であ

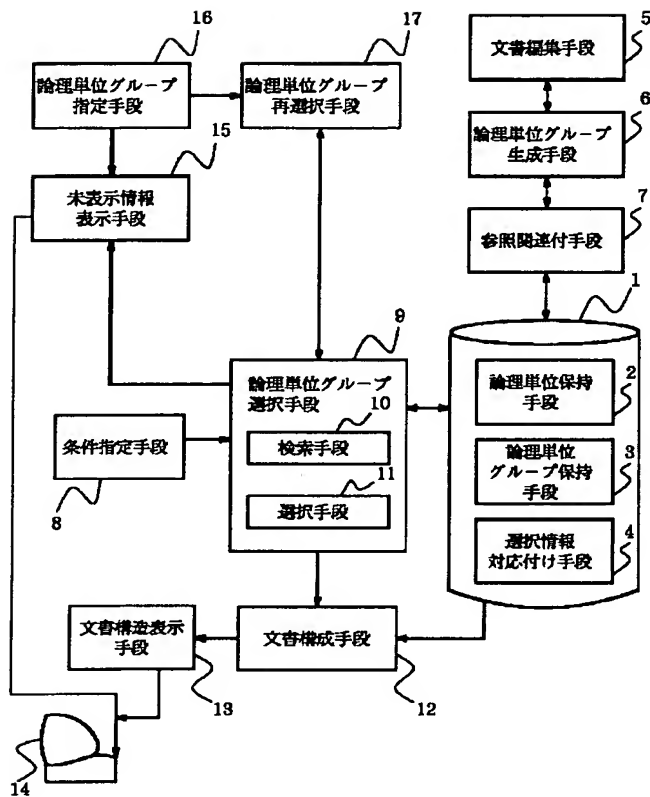
る。

- 【図52】 編集履歴の一例を示す図である。
- 【図53】 導出木構造の一例を示す図である。
- 【図54】 編集中の文書データ構造の一例を示す図である。
- 【図55】 ハイパー構造のデータ構造の一例を示す図である。
- 【図56】 ハイパー構造のデータ構造の一例を示す図である。
- 【図57】 ハイパー構造のデータ構造の一例を示す図である。
- 【図58】 新たに生成される文書データの一例を示す図である。
- 【図59】 履歴表示手段の画面表示の一例を示す図である。
- 【図60】 抽出指示手段の画面表示の一例を示す図である。
- 【図61】 編集手段の画面表示の一例を示す図である。
- 【図62】 抽出指示手段の画面表示の一例を示す図である。
- 【図63】 編集手段の画面表示の一例を示す図である。
- 【図64】 抽出指示手段の画面表示の一例を示す図である。
- 【図65】 編集手段の画面表示の一例を示す図である。
- 【図66】 抽出指示手段の画面表示の一例を示す図である。
- 【図67】 編集手段の画面表示の一例を示す図である。
- 【図68】 抽出指示手段の画面表示の一例を示す図である。
- 【図69】 編集手段の画面表示の一例を示す図である。
- 【図70】 抽出指示手段の画面表示の一例を示す図である。
- 【図71】 編集手段の画面表示の一例を示す図である。
- 【図72】 抽出指示手段の画面表示の一例を示す図である。
- 【図73】 編集手段の画面表示の一例を示す図である。
- 【図74】 編集手段の画面表示の一例を示す図である。
- 【図75】 カレント構造及びハイパー構造のデータ構造の一例を示す図である。

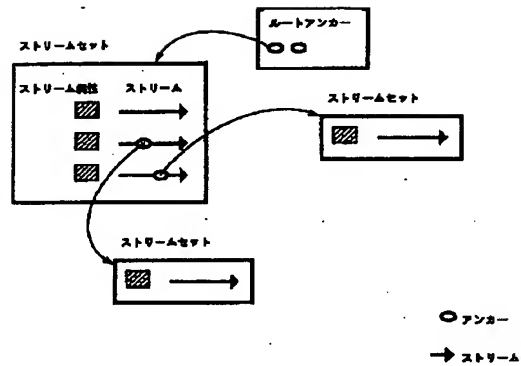
造の一例を示す図である。

- 【図76】 本発明の他の一実施例に係る図形の編集過程を示す図である。
  - 【図77】 図形編集によるデータ構造の一例を示す図である。
  - 【図78】 ハイパー構造のデータ構造の一例を示す図である。
  - 【図79】 抽出された図形データ構造の一例を示す図である。
  - 【図80】 本発明の更に他の一実施例に係るハイパーテキストの編集過程の一例を示す図である。
  - 【図81】 本発明の更に他の一実施例に係るハイパーテキストの編集過程の他の一例を示す図である。
  - 【図82】 ハイパーテキストの履歴の一例を示す図である。
  - 【図83】 ハイパーテキストの抽出処理を説明する図である。
  - 【図84】 従来例に係る文書構造を示す概念図である。
  - 【図85】 内容部が変更された従来例の文書構造を示す概念図である。
  - 【図86】 部分構造が挿入された従来例の文書構造を示す概念図である。
  - 【図87】 部分構造が削除された従来例の文書構造を示す概念図である。
  - 【図88】 部分構造の順所が入れ替えられた従来例の文書構造を示す概念図である。
  - 【図89】 部分構造が移動された従来例の文書構造を示す概念図である。
- 【符号の説明】
- 1・・・メモリ、 2・・・論理単位保持手段、 3・・・論理単位グループ保持手段、 4・・・選択情報対応付け手段、 5・・・文書編集手段、 6・・・論理単位グループ生成手段、 7・・・参照関連付手段、 8・・・条件指定手段、 9・・・論理単位グループ選択手段、 12・・・文書構成手段、 13・・・文書構造表示手段、 14・・・ディスプレイ、 15・・・未表示情報表示手段、 16・・・論理単位グループ指定手段、 17・・・論理単位グループ再選択手段、 20・・・ハイパー構造管理手段、 21・・・カレント構造管理手段、 22・・・編集手段、 24・・・登録手段、 25・・・履歴管理手段、 27・・・抽出手段、 28・・・抽出指示手段、 33・・・検索手段、 34・・・選択手段、 35・・・検索条件指示手段、 36・・・選択規則指示手段、

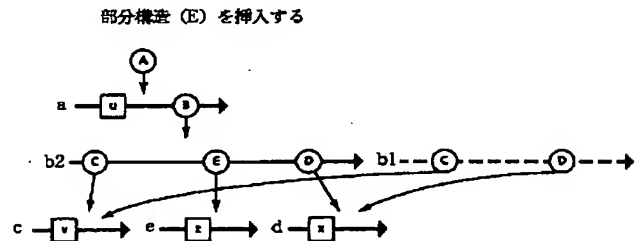
【図1】



【図2】

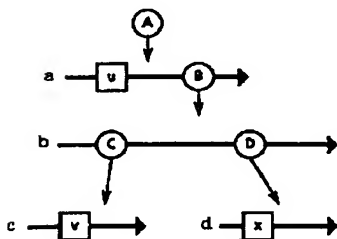


【図5】

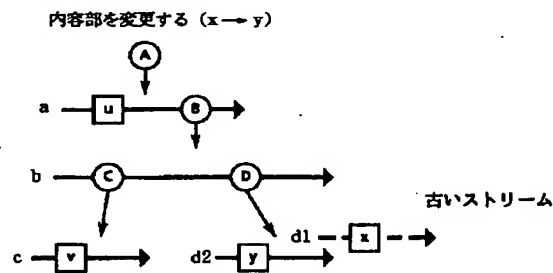


【図11】

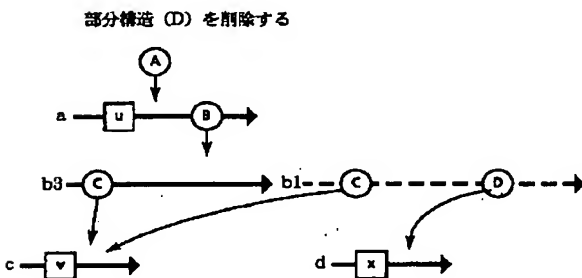
【図3】



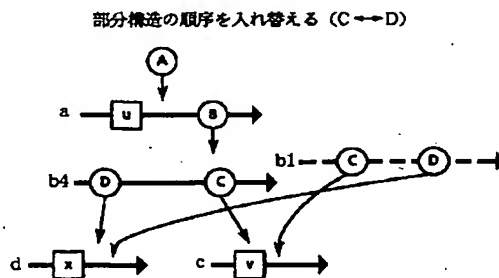
【図4】



【図6】



【図7】



新規文書の名前を入力して下さい

文書ルート: 発明説明書

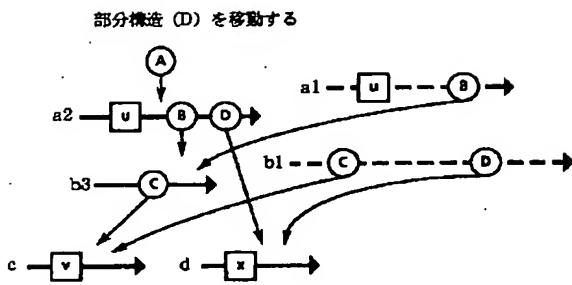
ok

日時: 1995.1.1 15:00

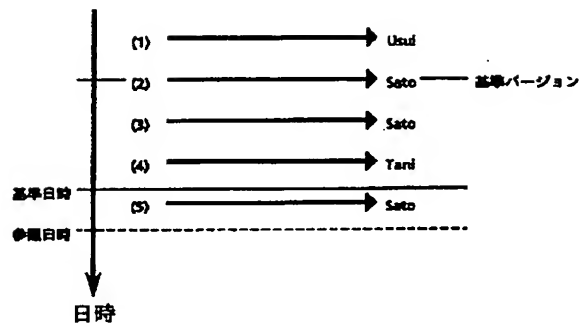
ユーザ: Sato

BEST AVAILABLE COPY

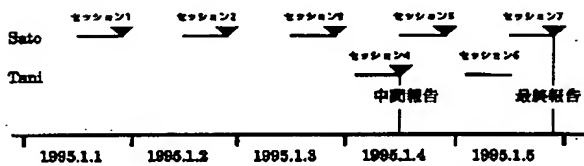
【図8】



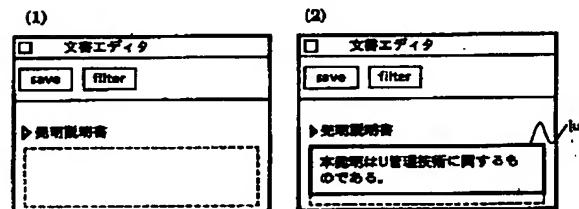
【図9】



【図10】



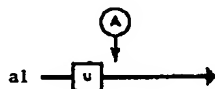
【図12】



【図13】

文書構造

【図14】



必要な設定をして下さい

文書ルート: 発明説明書

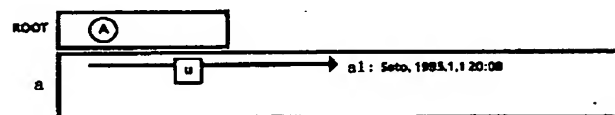
ok

バージョン設定:

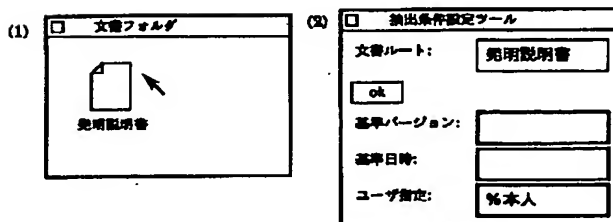
日時: 1995.1.1 20:00

ユーザ: Sato

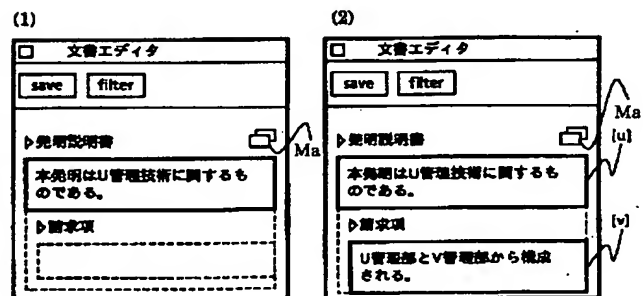
【図15】



【図16】

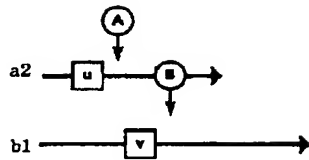


【図17】



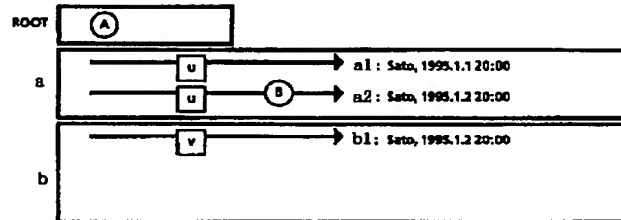
【図18】

文書構造



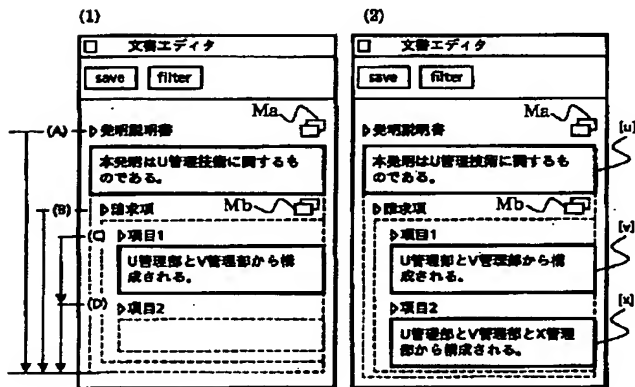
【図19】

複合文書構造



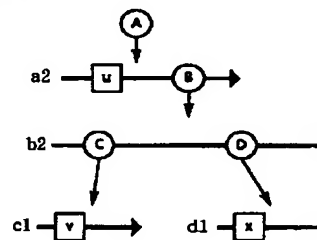
【図20】

【図21】



【図22】

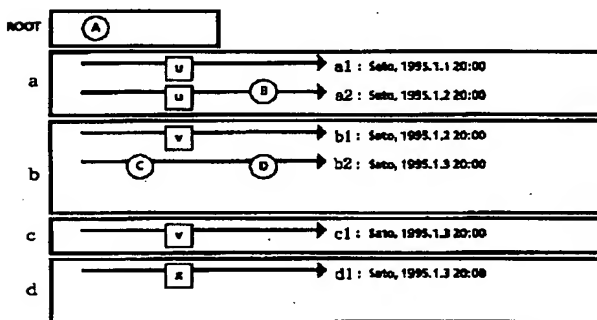
文書構造



【図23】

REST AVAILABLE COPY

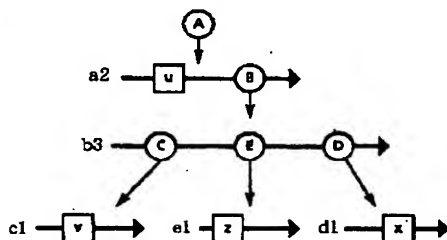
複合文書構造



【図25】

【図26】

文書構造



必要な設定をして下さい

文書ルート:

ok

バージョン設定:

日時:

ユーザ:

抽出条件設定ツール

文書ルート:

ok

基準バージョン:

基準日時:

ユーザ指定:

【図35】

必要な設定をして下さい

文書ルート:

ok

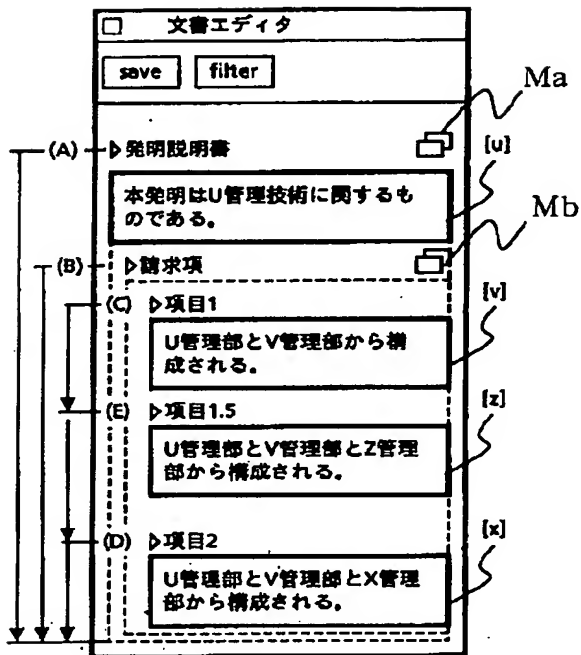
バージョン設定:

日時:

ユーザ:

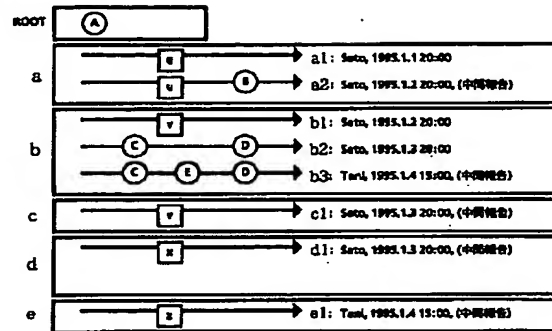


【図24】

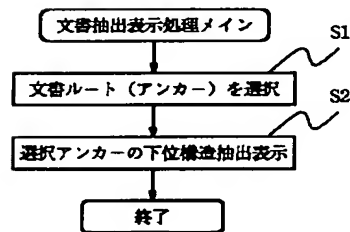


【図27】

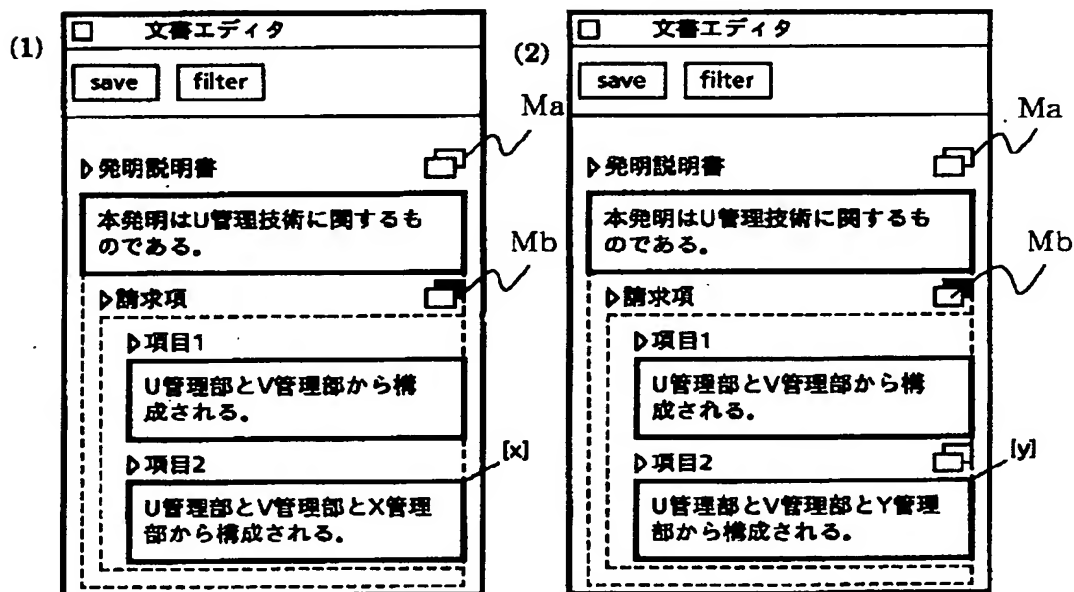
複合文書構造



【図37】



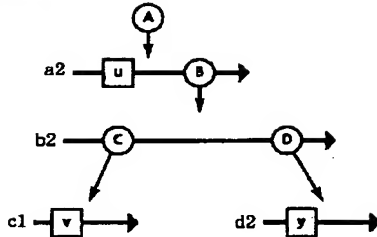
【図28】



BEST AVAILABLE COPY

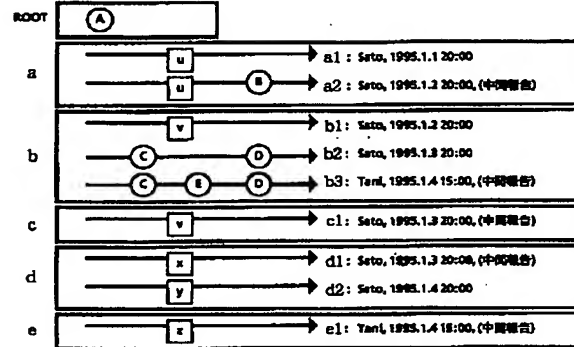
【図29】

## 文書構造



【図30】

## 複合文書構造



【図31】

【図33】

## (1) 文書エディタ起動

☐ 抽出条件設定ツール

文書ルート: 発明説明書

ok

基準バージョン: 中間報告

基準日時:

ユーザ指定: %本人

## (2) 抽出条件変更

☐ 抽出条件設定ツール

文書ルート: 発明説明書

ok

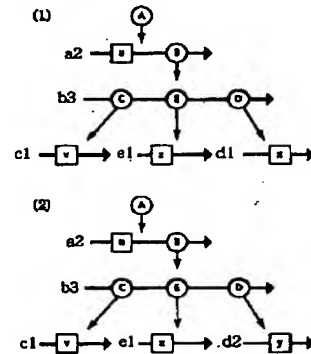
基準バージョン: 中間報告

基準日時:

ユーザ指定:

【図32】

## 文書構造



(1) 文書エディタ

save filter

▶ 発明説明書

本発明はU管理技術に関するものである。

▶ 請求項

▶ 項目1

U管理部とV管理部から構成される。

▶ 項目1.5

U管理部とV管理部とZ管理部から構成される。

▶ 項目2

U管理部とV管理部とX管理部から構成される。

(2) 文書エディタ

save filter

▶ 発明説明書

本発明はU管理技術に関するものである。

▶ 請求項

▶ 項目1

U管理部とV管理部から構成される。

▶ 項目1.5

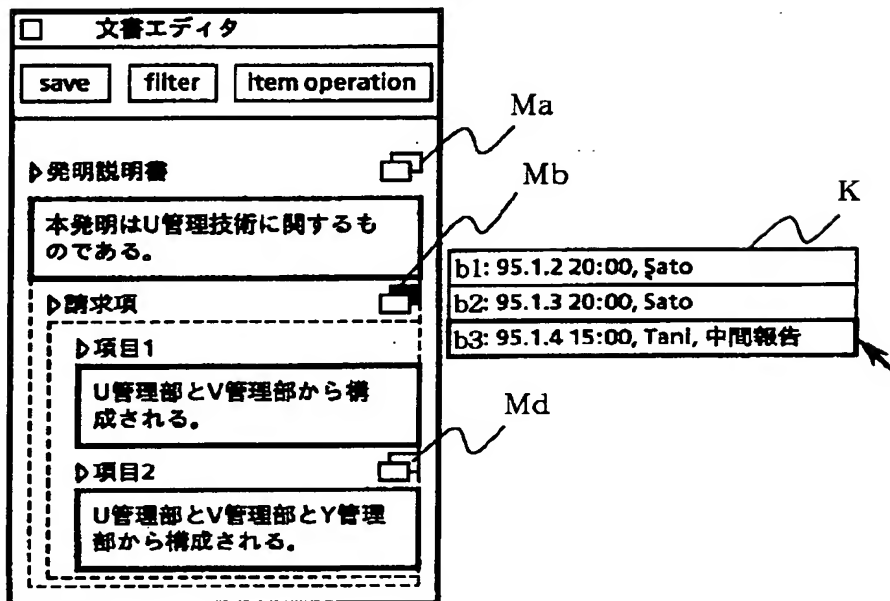
U管理部とV管理部とZ管理部から構成される。

▶ 項目2

U管理部とV管理部とY管理部から構成される。

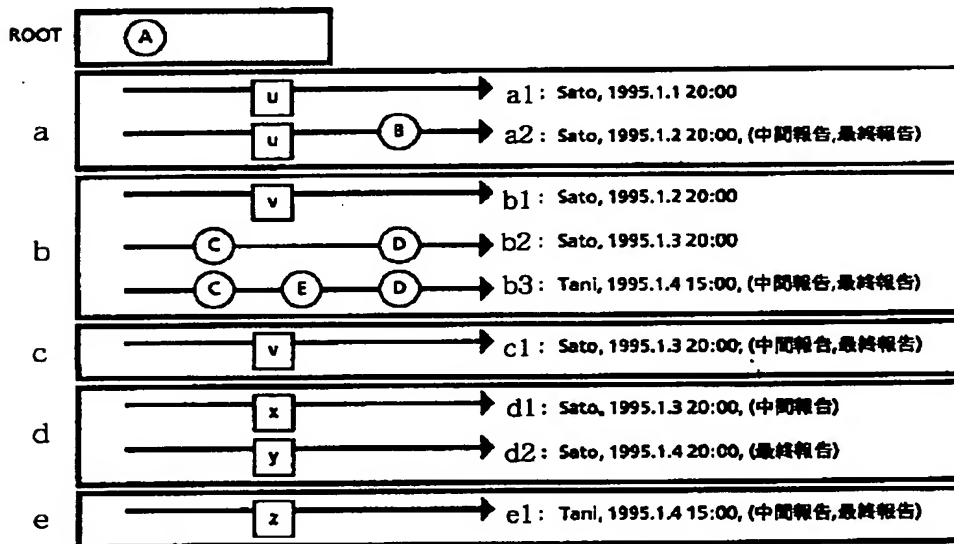
BEST AVAILABLE COPY

【図 3 4】

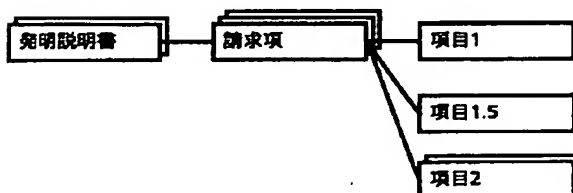


【図 3 6】

## 複合文書構造

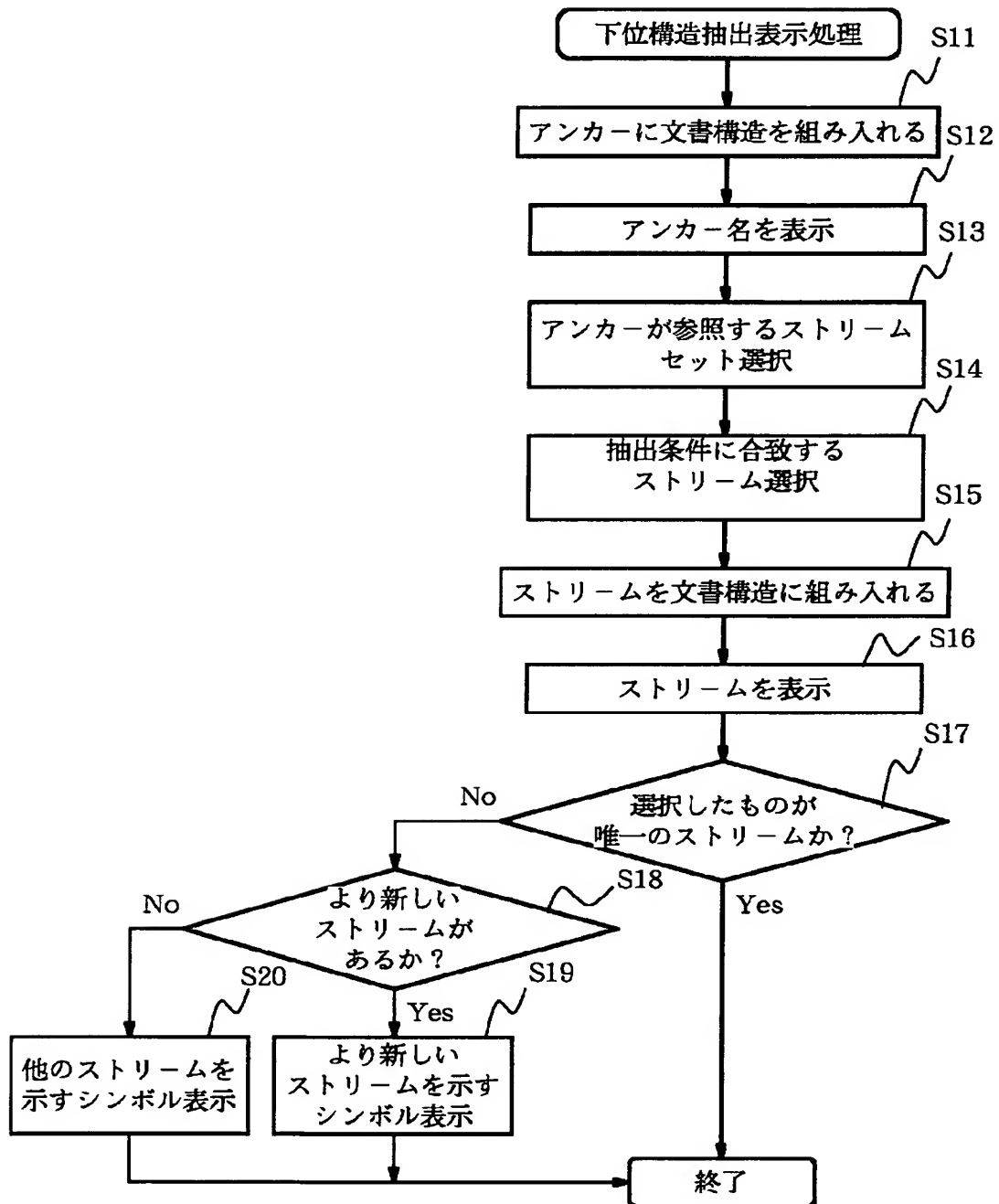


【図 4 4】



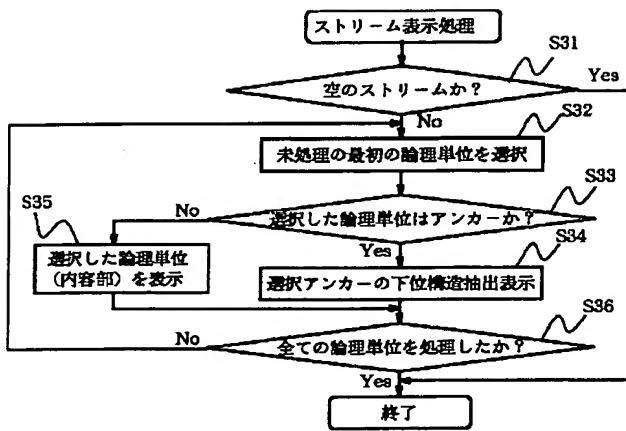
BEST AVAILABLE COPY

【図38】

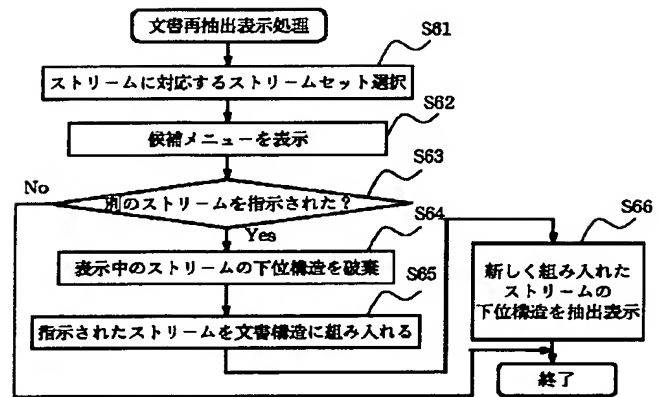


BEST AVAILABLE COPY

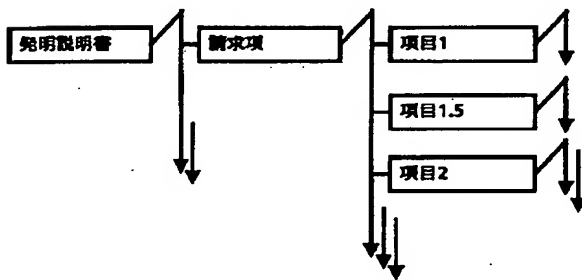
【図39】



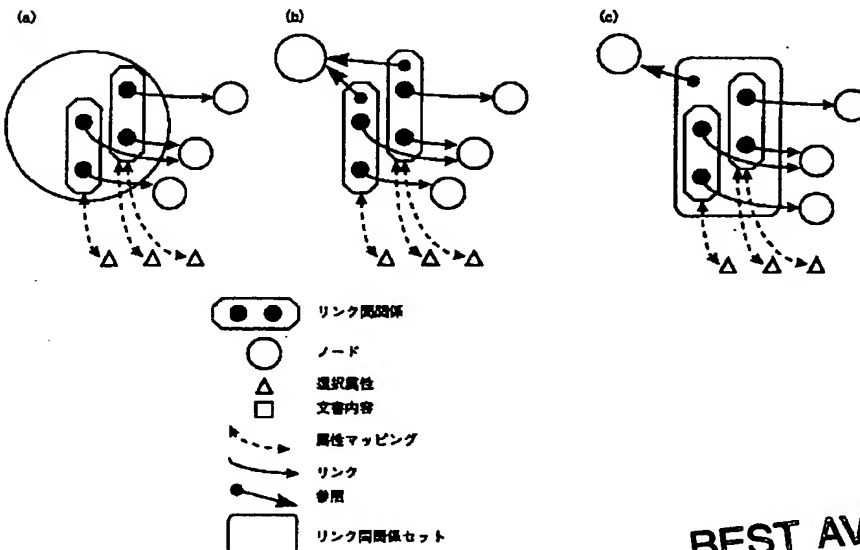
【図41】



【図43】

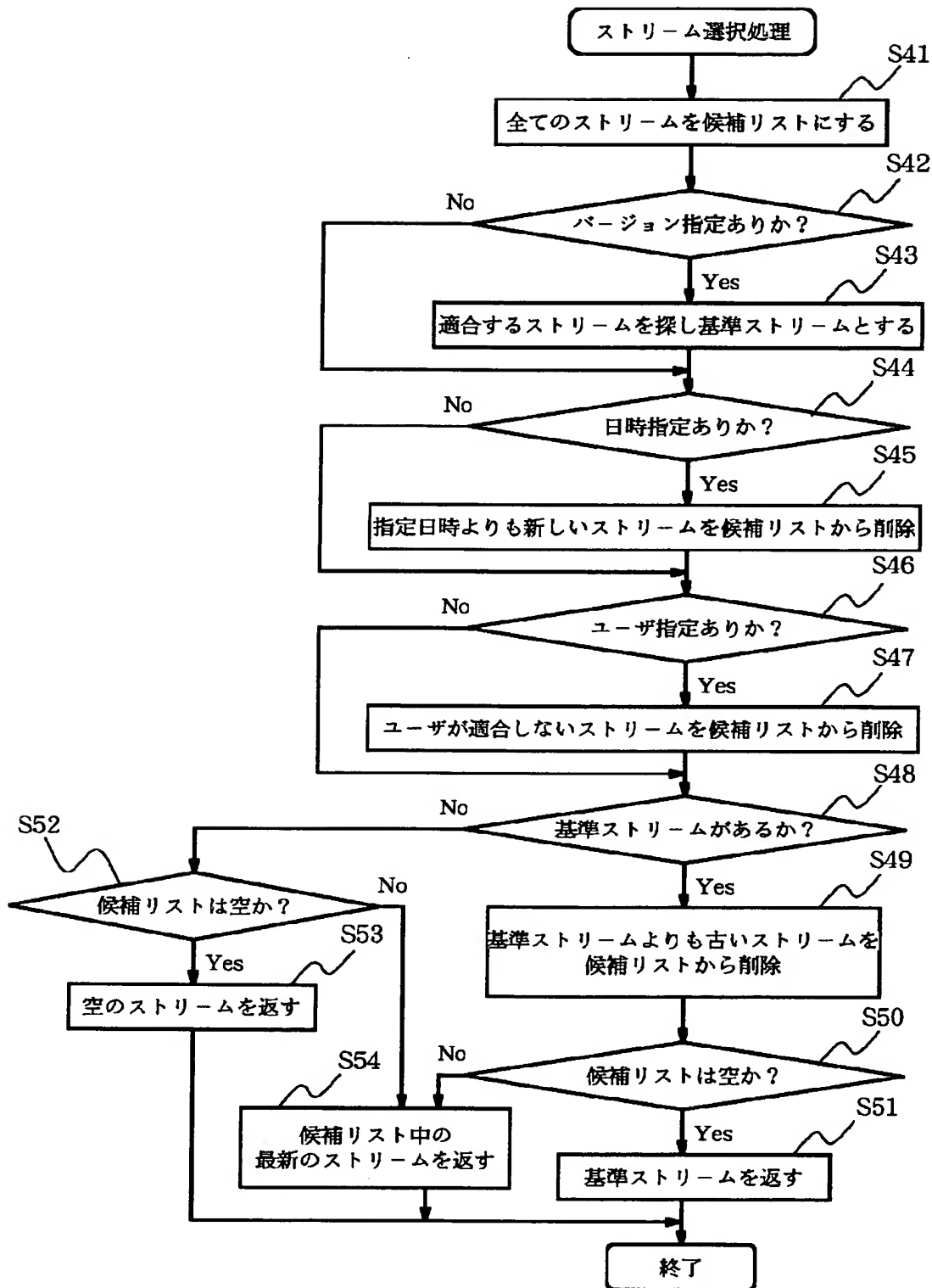


【図45】



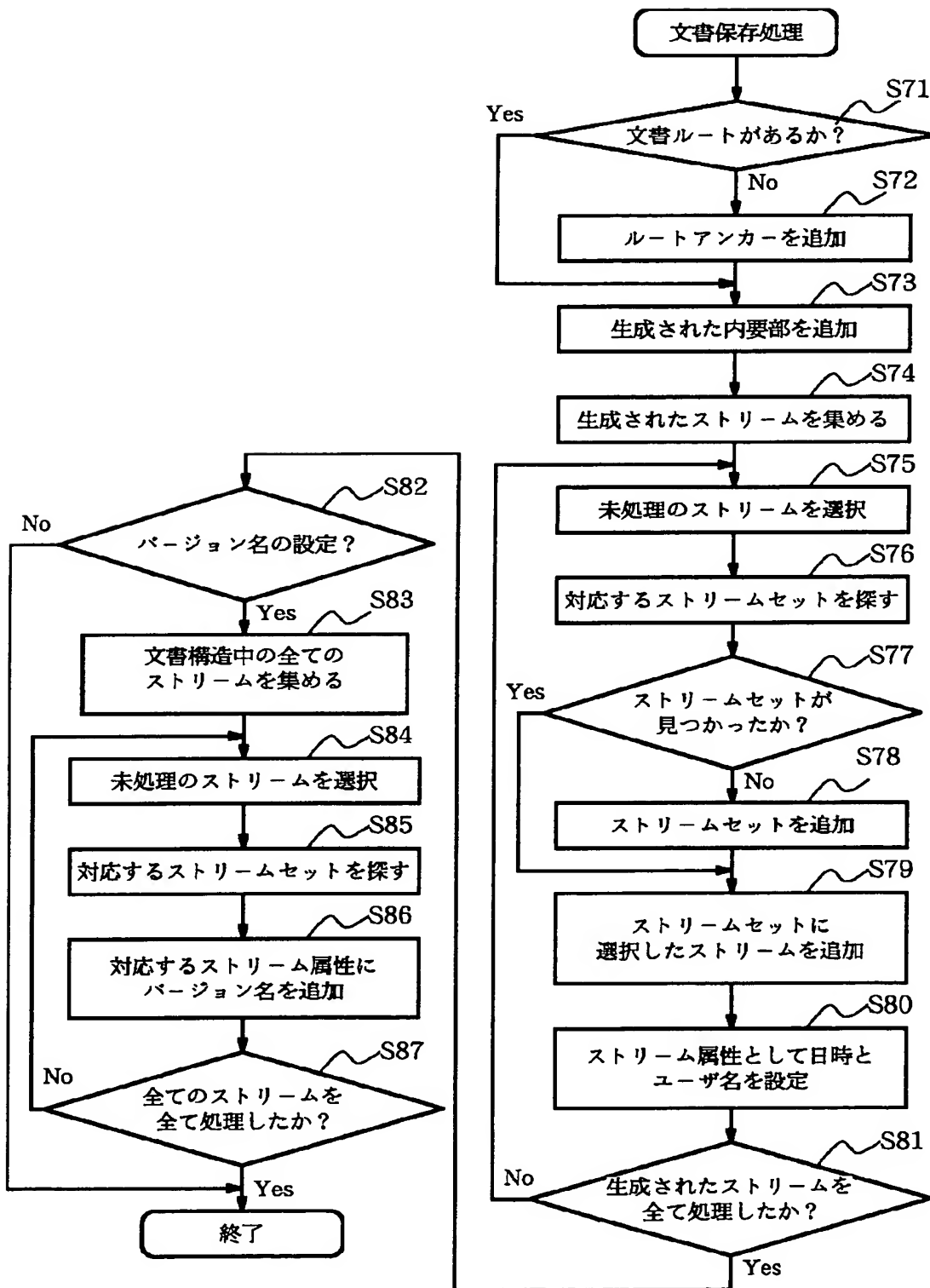
BEST AVAILABLE COPY

【図40】

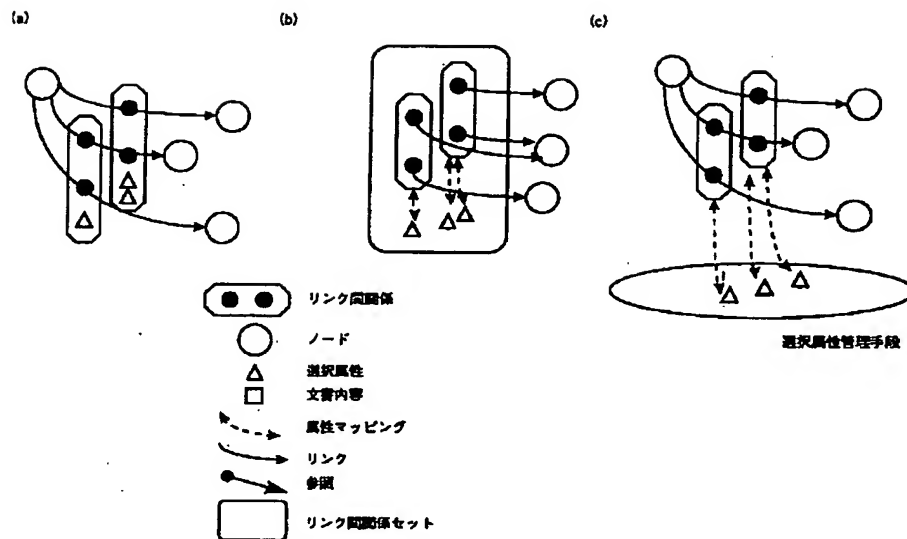


BEST AVAILABLE COPY

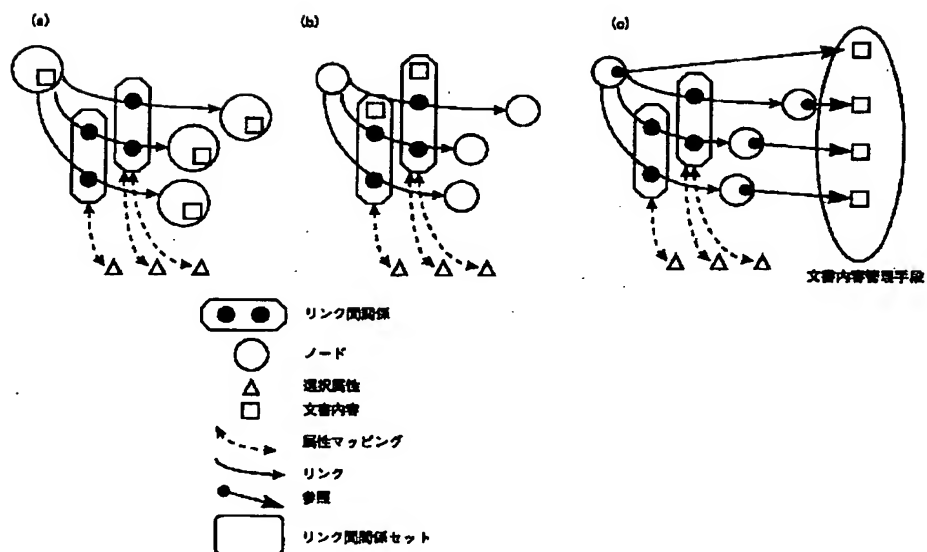
【図42】



【図46】



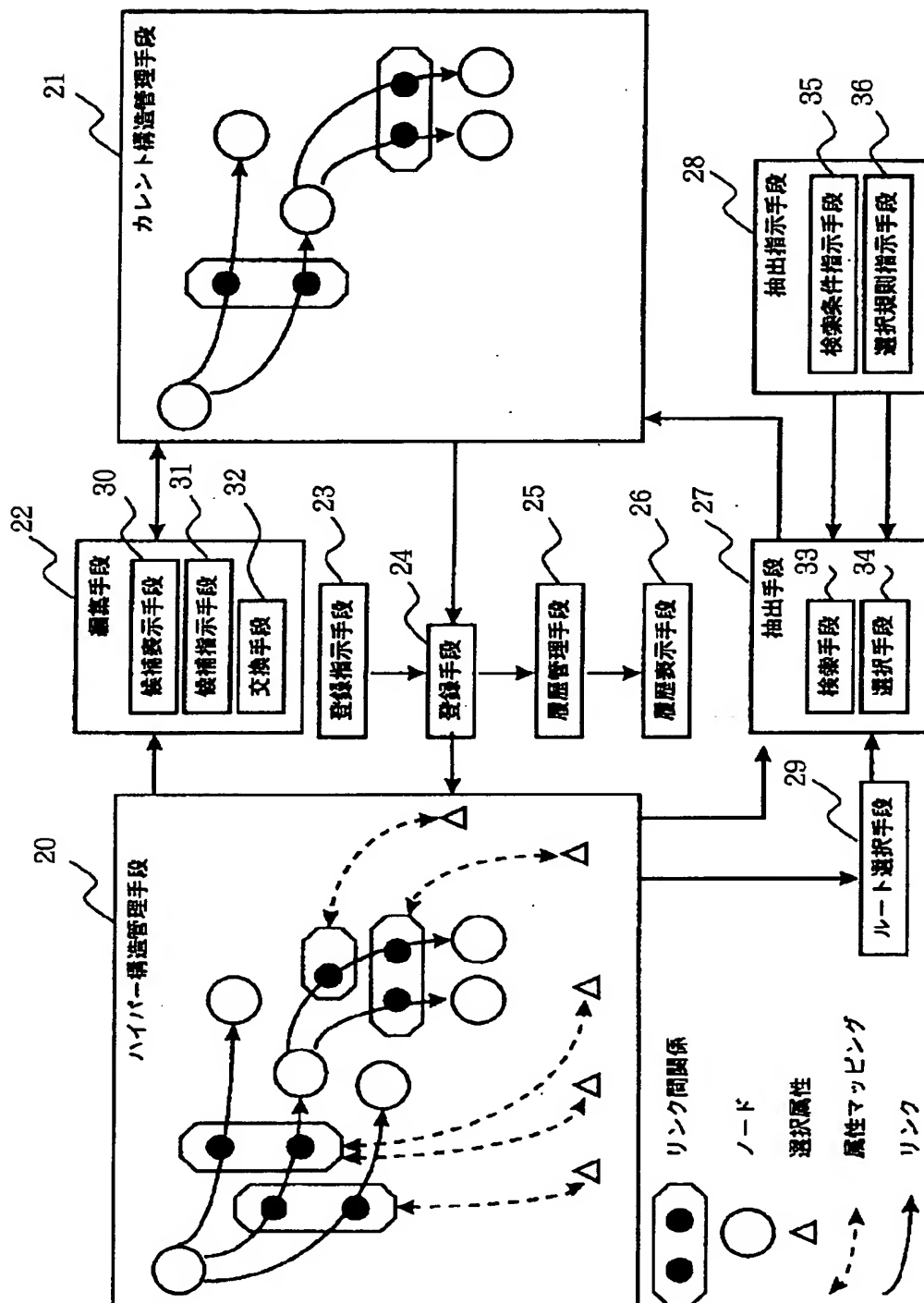
【図47】



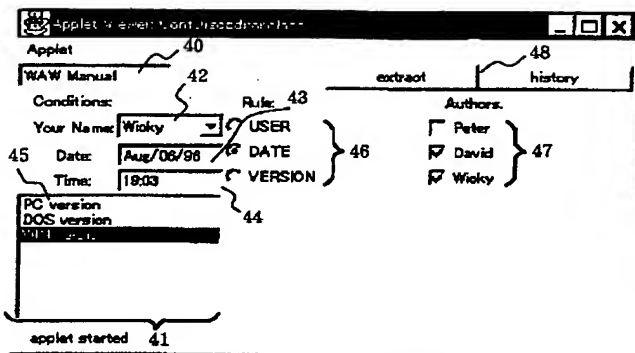
REST AVAILABLE COPY



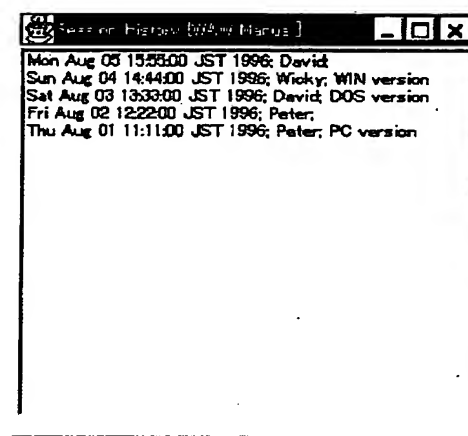
【図48】



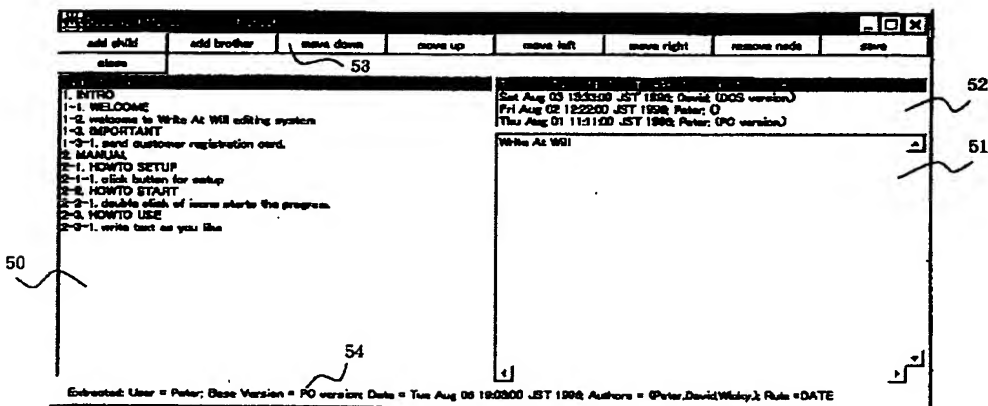
【図49】



【図50】

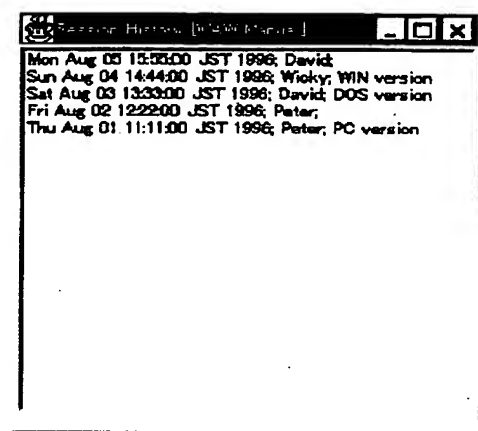
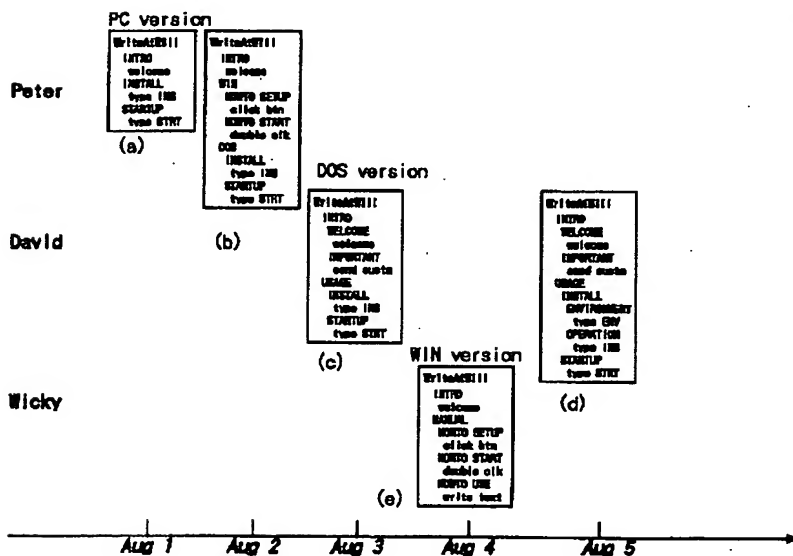


【図51】



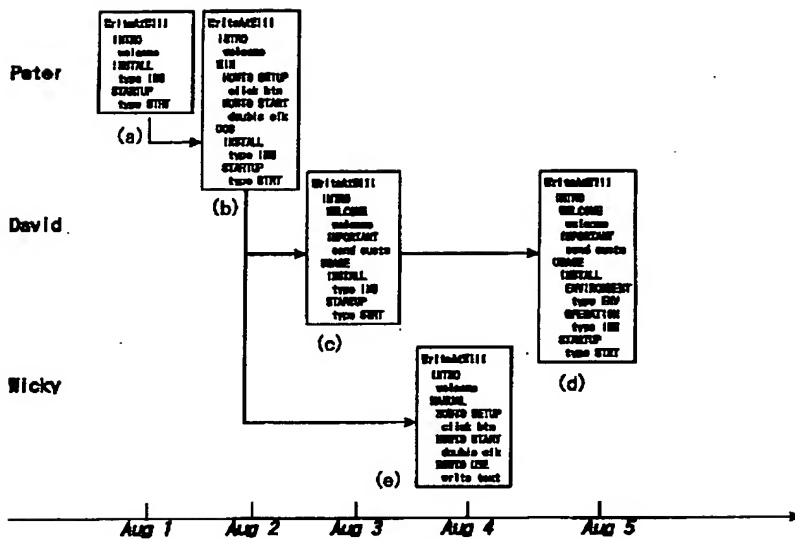
【図52】

【図59】

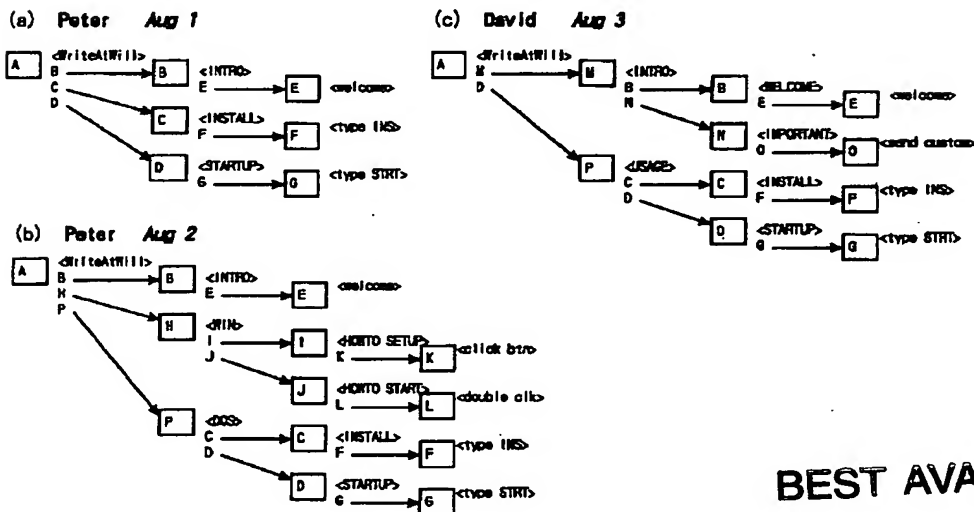


BEST AVAILABLE COPY

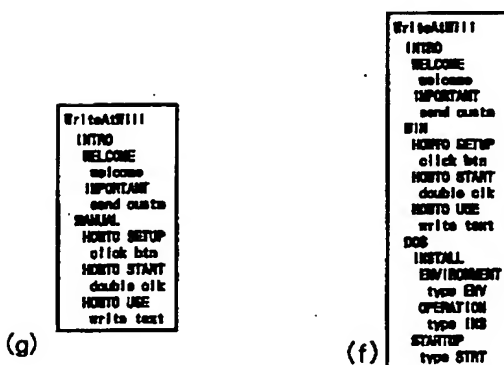
【図53】



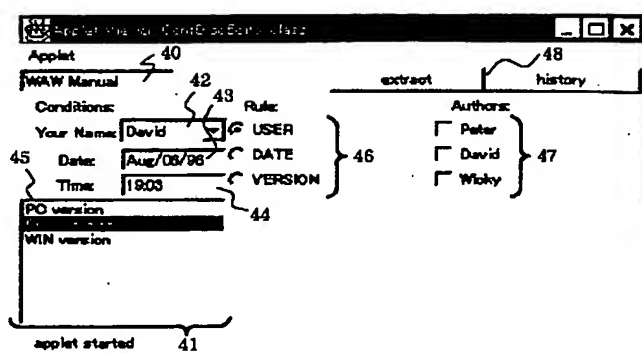
【図54】



【図58】

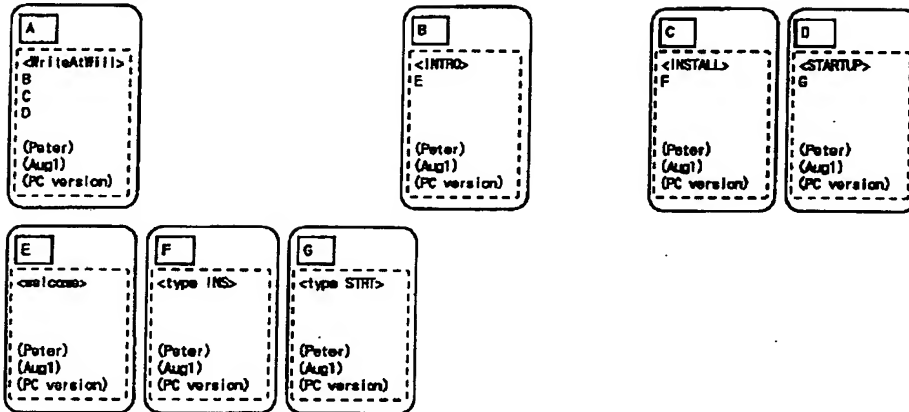


【図60】

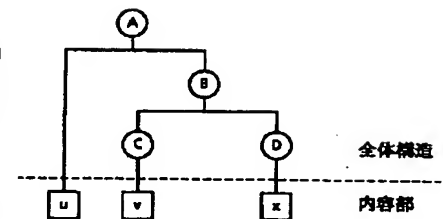
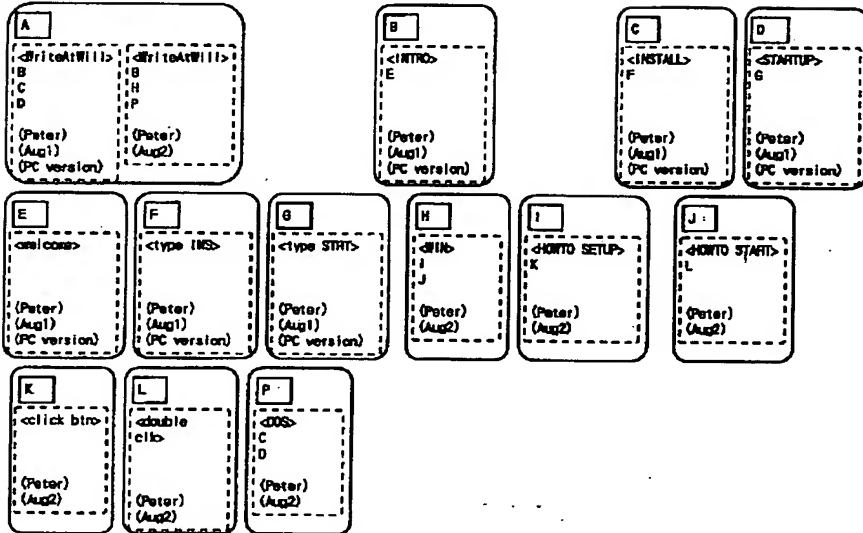


BEST AVAILABLE COPY

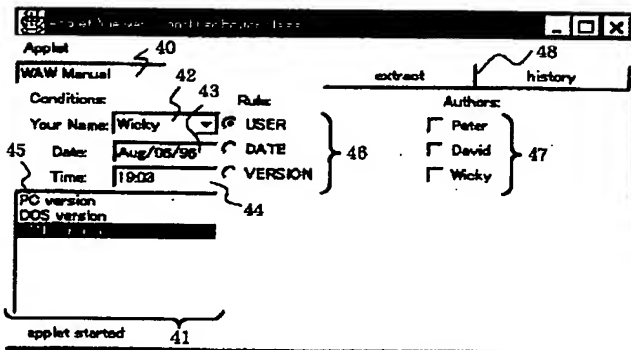
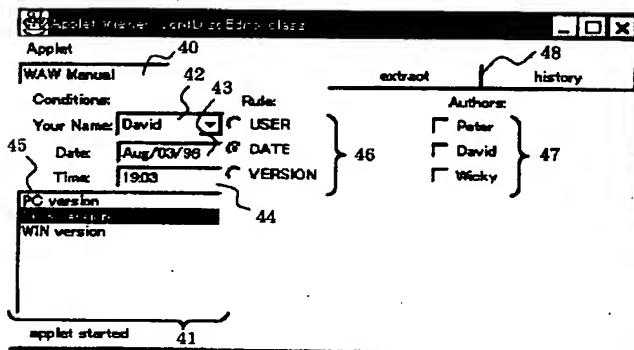
**Peter Aug 1**



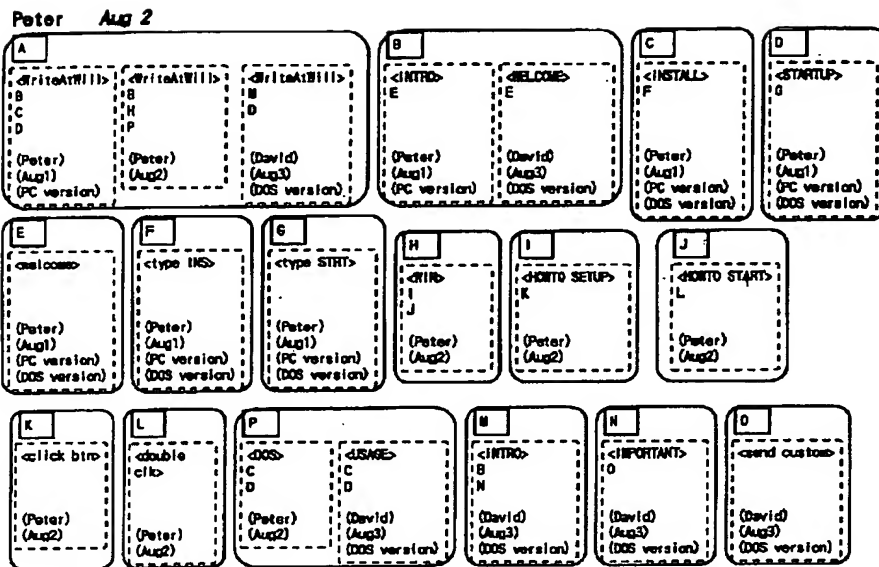
【図84】



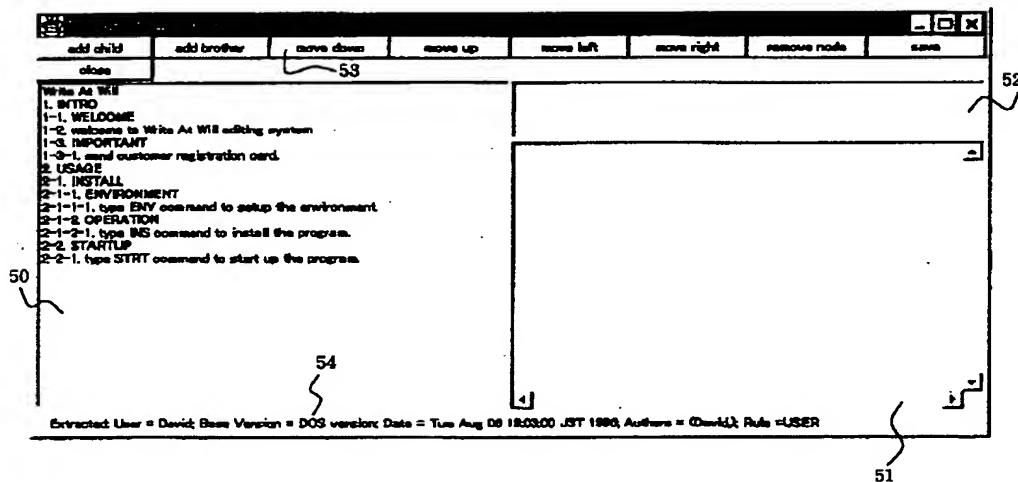
【图64】



【図57】

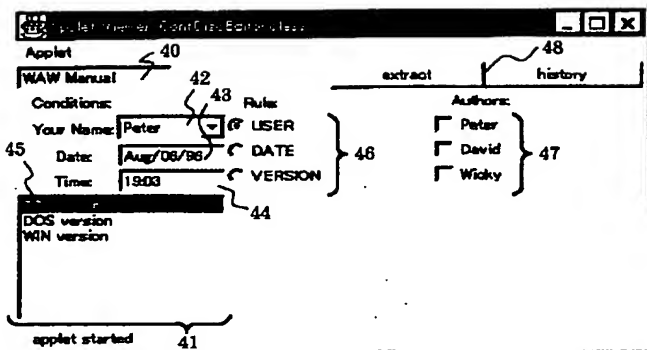
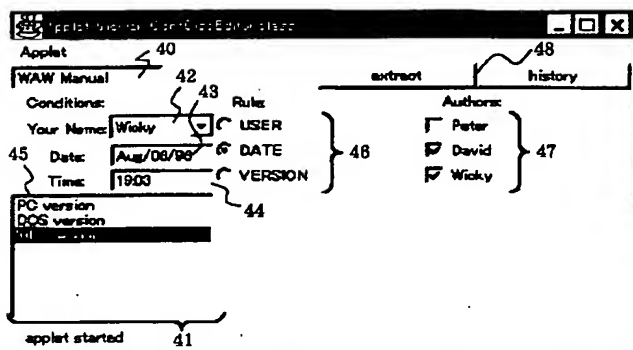


【図61】



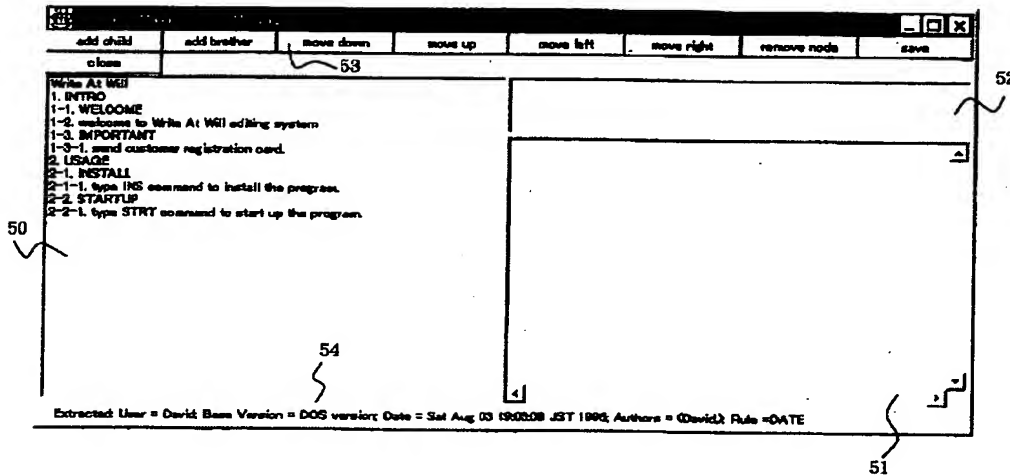
【図66】

【図68】

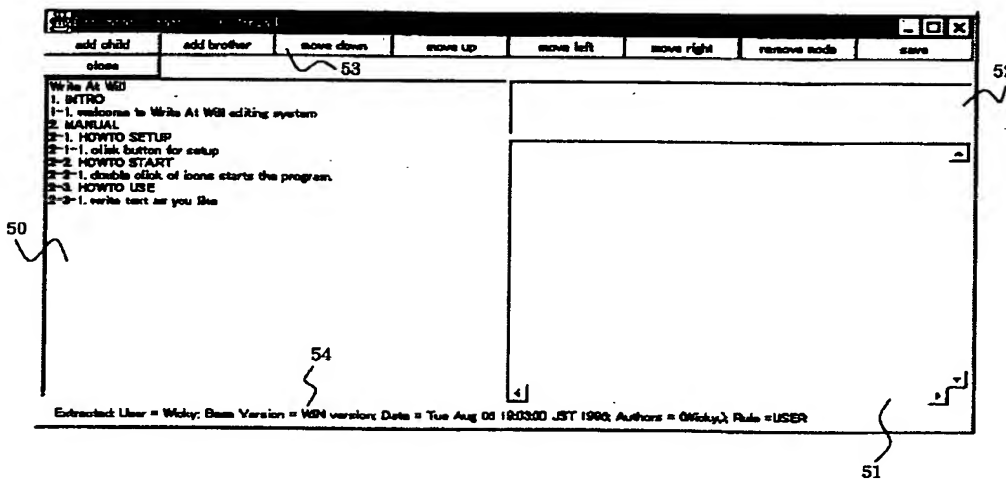


BEST AVAILABLE COPY

【図63】

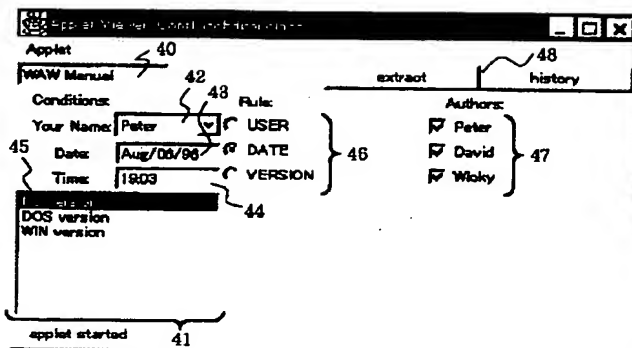
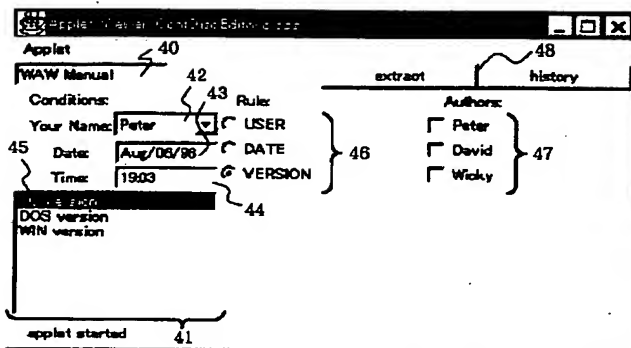


【図65】



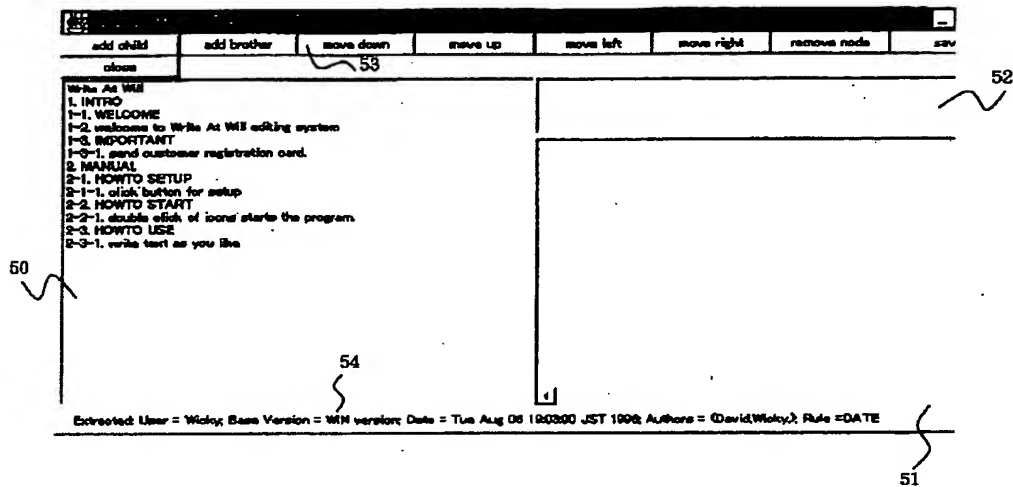
【図70】

【図72】

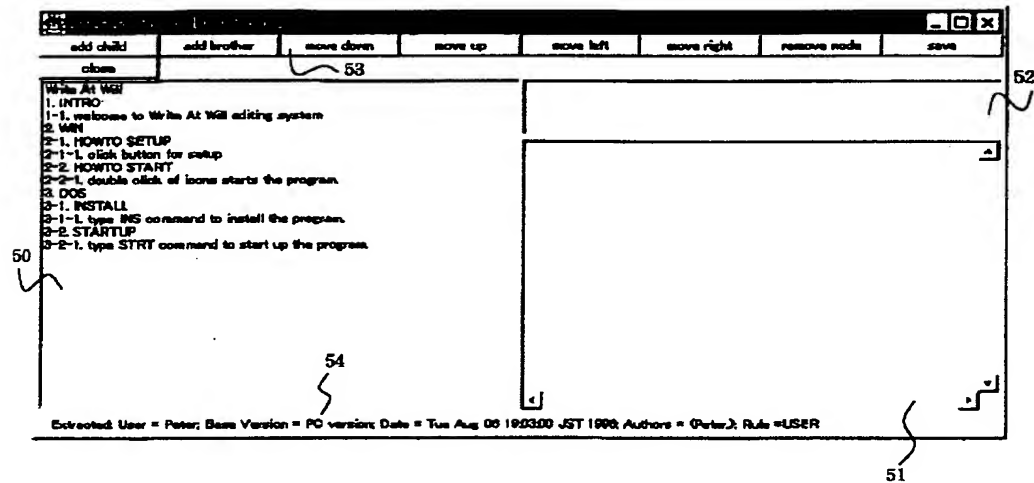


REST AVAILABLE COPY

【図67】

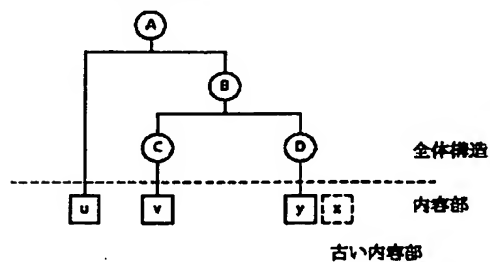


【図69】



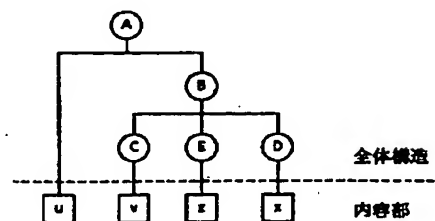
【図85】

内容部を変更する (x→y)



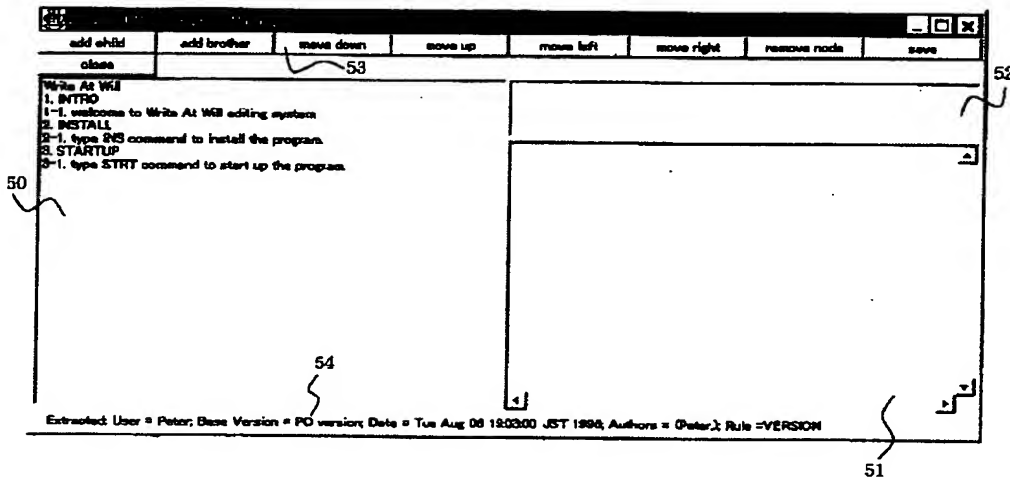
【図86】

部分構造(E)を挿入する

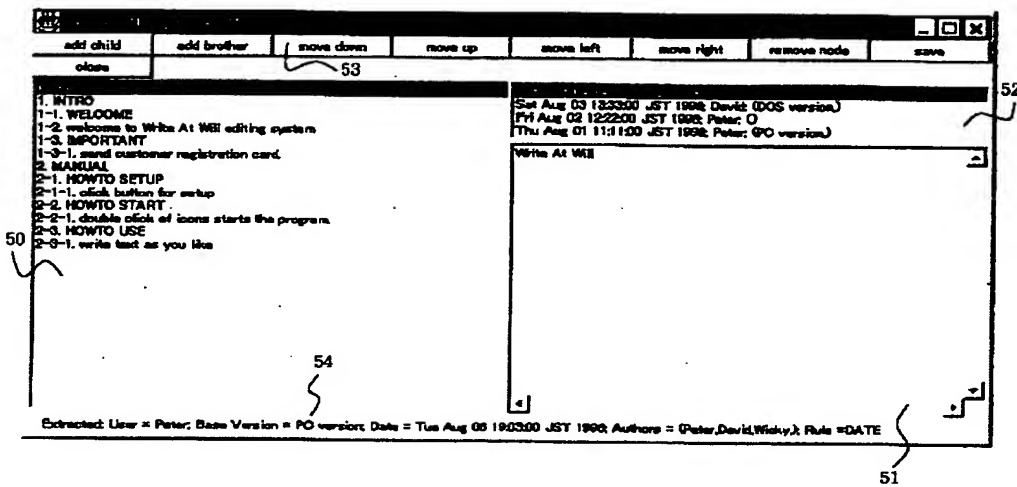


BEST AVAILABLE COPY

【図71】

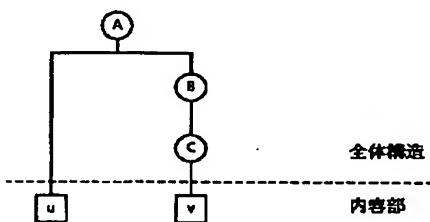


【図73】



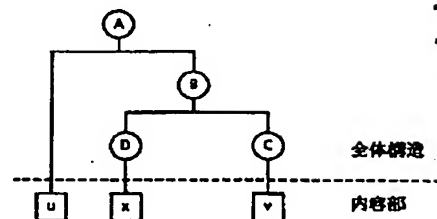
【図87】

部分構造(D)を削除する



【図88】

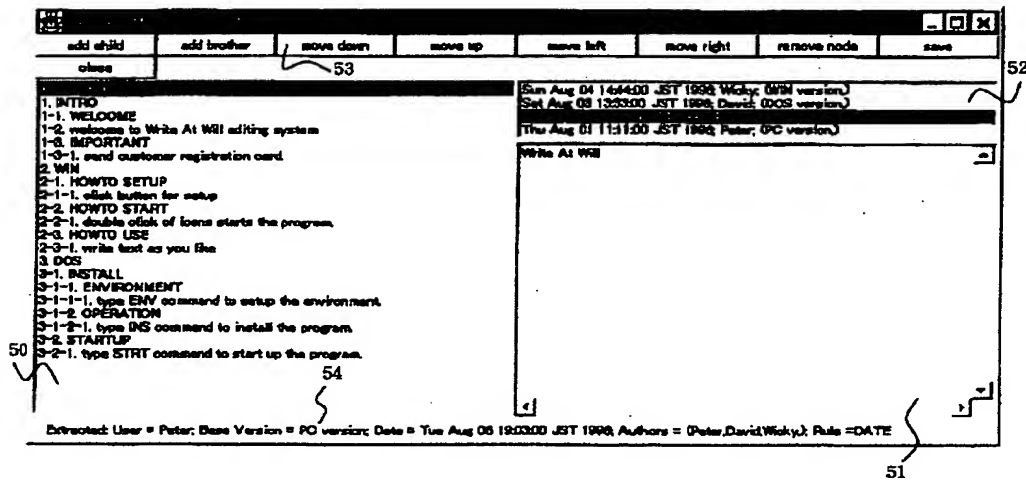
部分構造の順序を入れ替える(C&lt;=&gt;D)



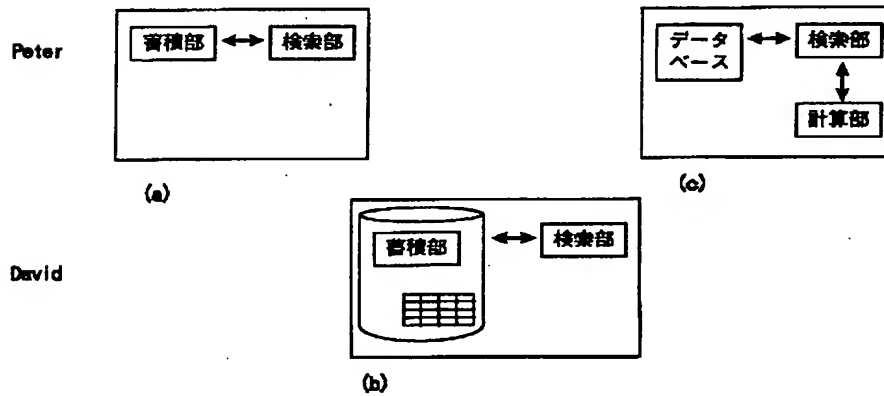
BEST AVAILABLE COPY



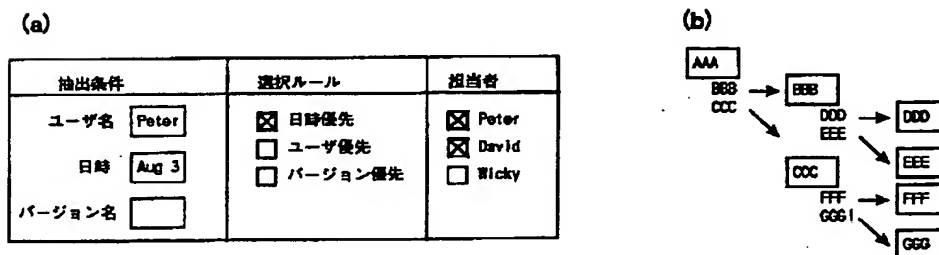
【図74】



【図76】

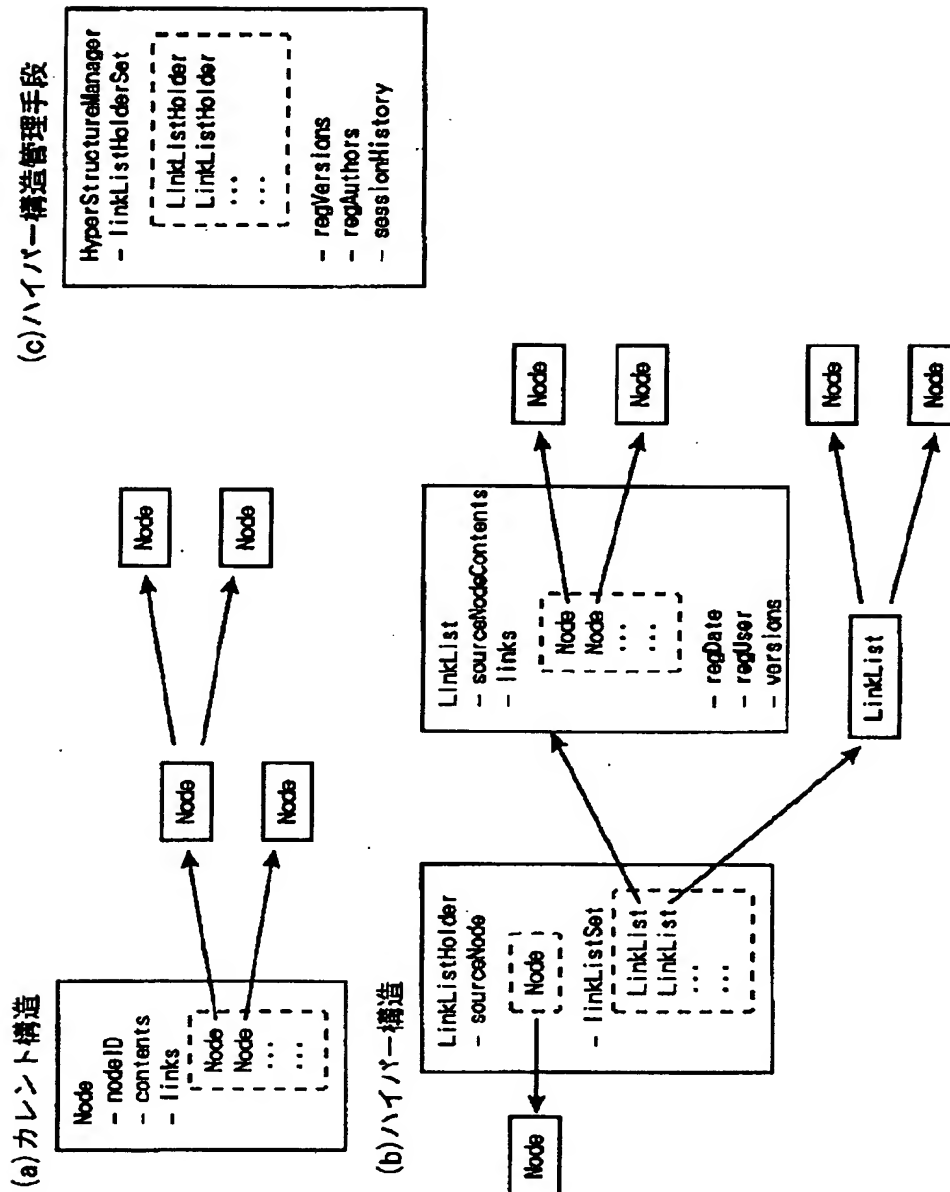


【図83】



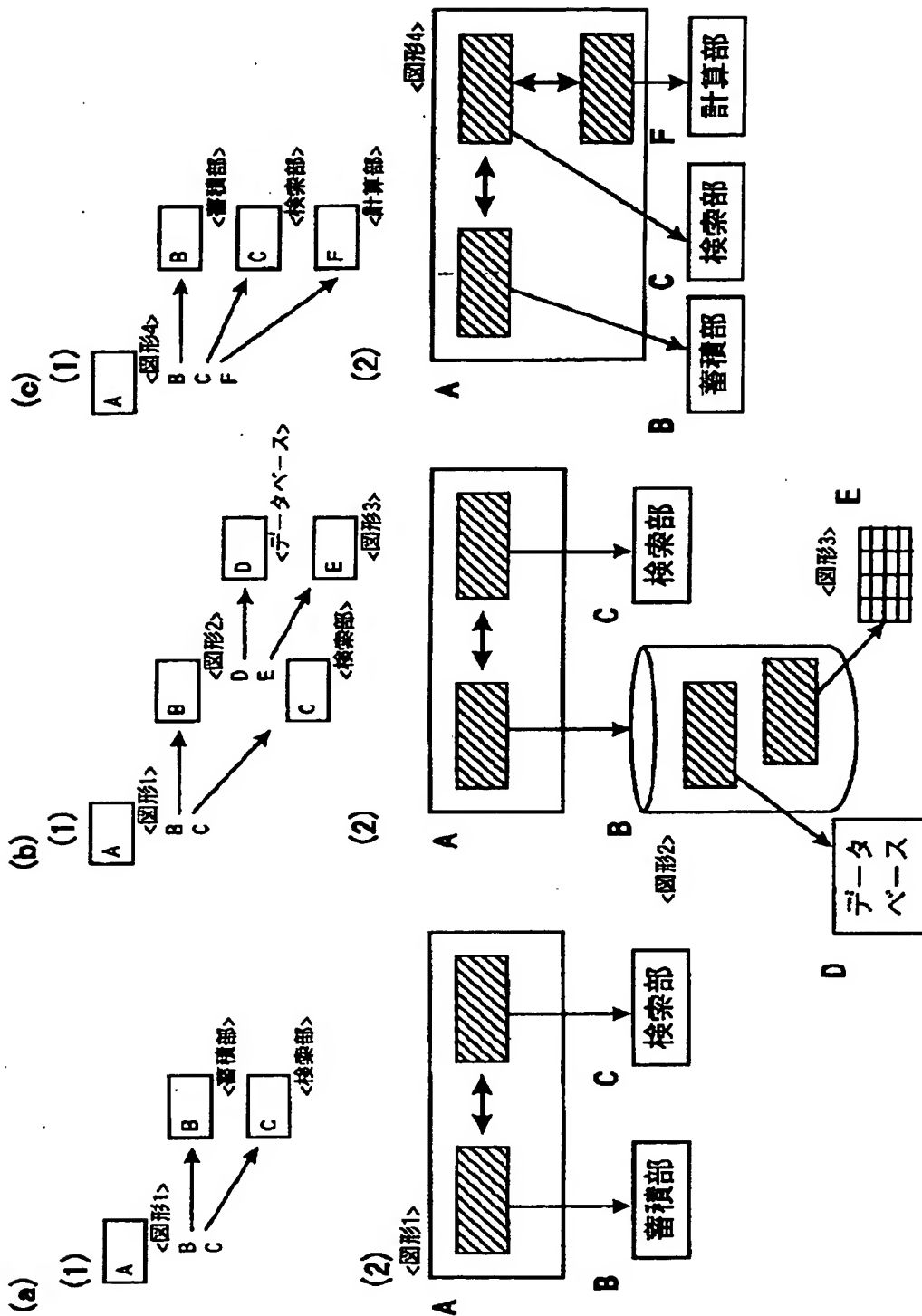
BEST AVAILABLE COPY

【図75】



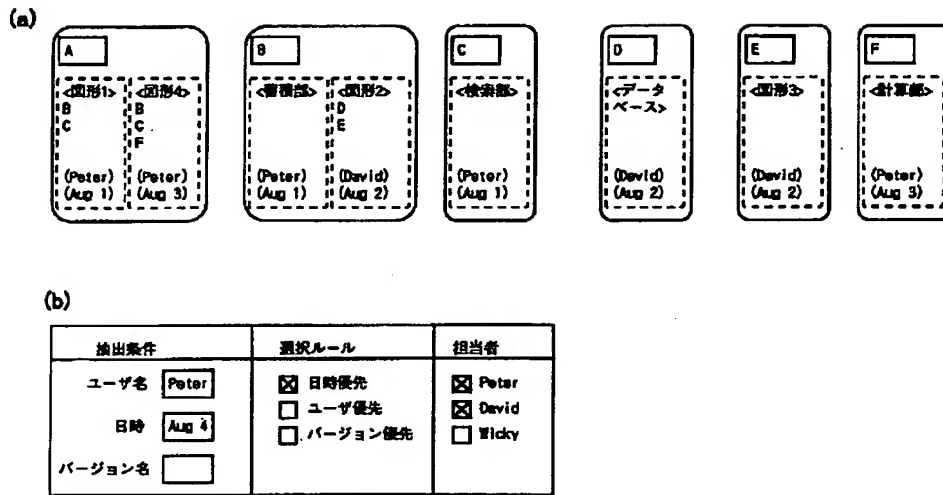
BEST AVAILABLE COPY

【図77】

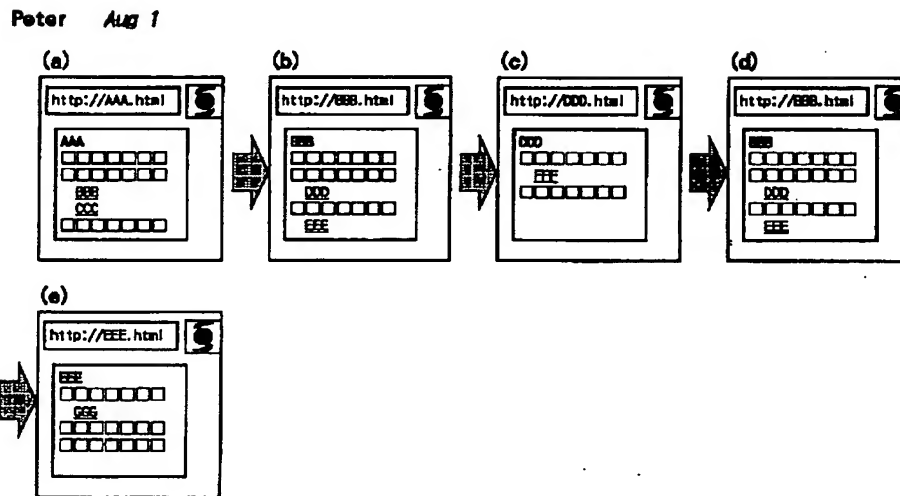


BEST AVAILABLE COPY

【図78】

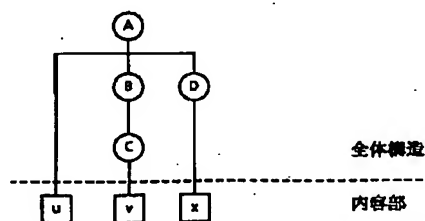


【図80】



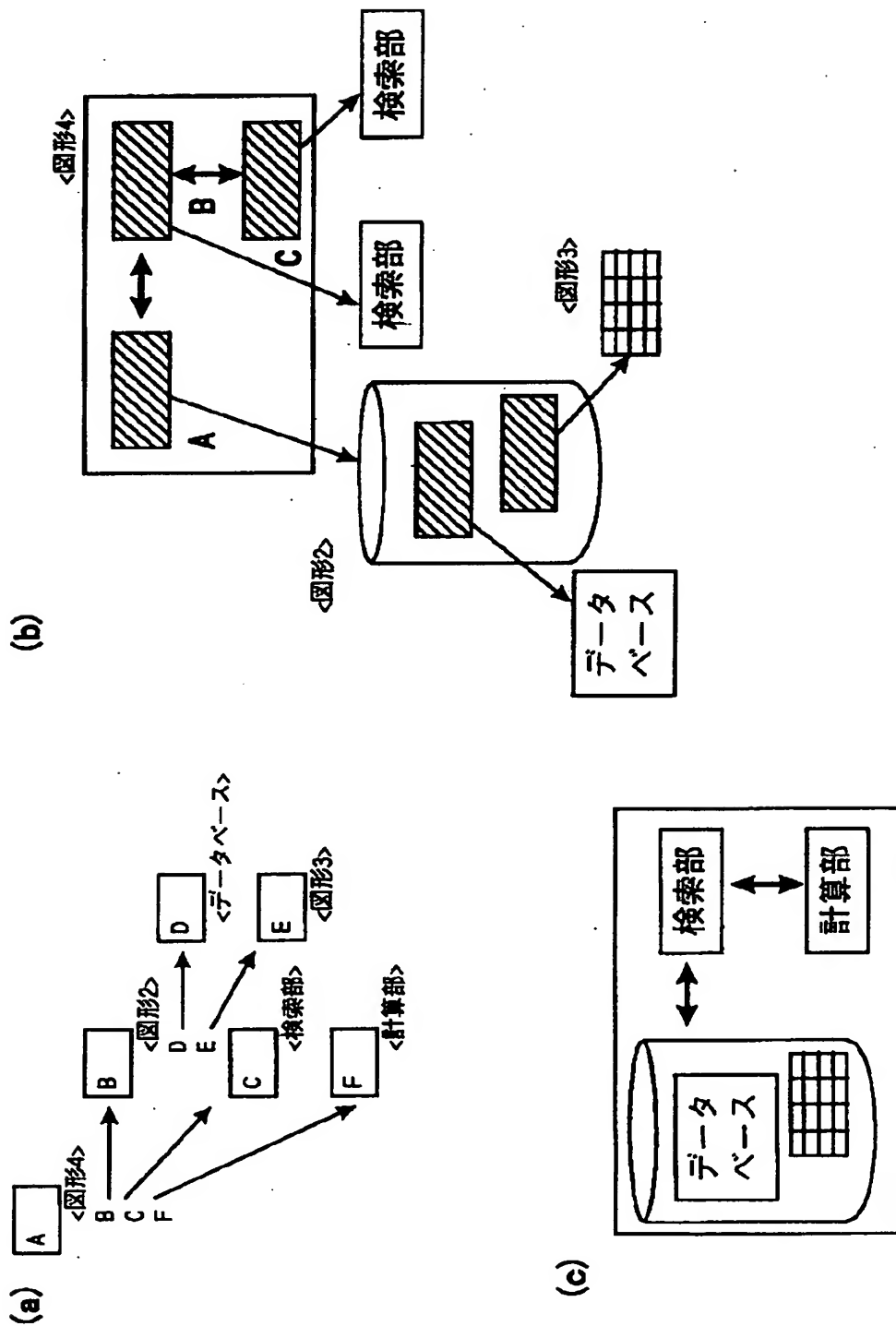
【図89】

部分構造(D)を移動する



BEST AVAILABLE COPY

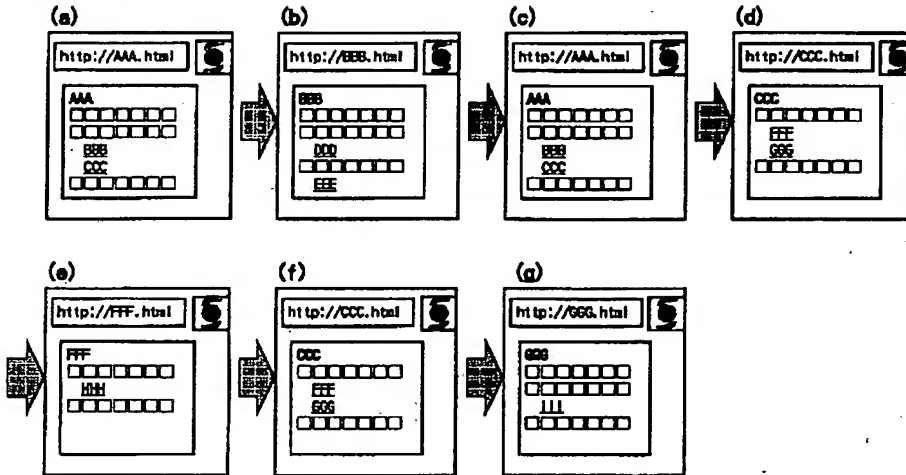
【図79】



BEST AVAILABLE COPY

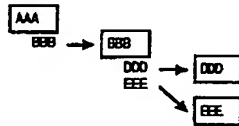
【図81】

David Aug 2

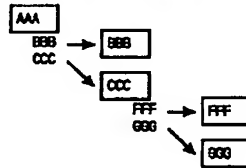


【図82】

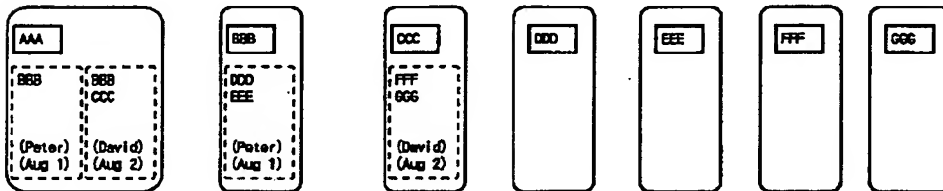
(a) Peter Aug 1



(b) David Aug 2



(c)



BEST AVAILABLE COPY

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**